*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```

*** Why using struct?
   -- 考慮寫一個 C/C++ 程式處理學生的成績資料, 例如輸入檔案如下

學號	NAME	身高	體重	S1	S2	S3	S4	S5
9817001	Chang-3-fong	175	70.5	85	78	92	66	83
9817003	Lee-4-dee	159	66	75	83	88	77	78
9817005	A Ban	163	56	83	87	93	83	84
9817007	D.H. Lee	182	85.4	82	75	90	76	78
9817011	Michael Chang	157	68.5	89	74	85	69	86
9817014	Hei Hei Hei	165	57.5	78	79	86	77	82

    算出每個人五科平均分數, 輸出要包括平均以及名次
   印兩次: 依照學號由小到大排印一次? 再依照分數高低排(即第一名先印)?

 思考? 如何存放資料? 很多 array? 一個學生的資料散置在許多 array 中?
     還是用 一個 array, 但是每一個元素是一個學生的所有資料 ? (better)
      ==> 用 struct Student {
               ...
               ...
             }; // 自訂資料結構

             struct Student x[999];

*** Why using class ?

     文法上:
        struct Hahaha {private:
         ...
        };
             相當於 class Hahaha 如右   class Hahaha {
                                                ...
                                              };
    意義上:
      把 data 以及對該些data 會做事的動作或方法(即函數)封藏在 class 內部

        ==> Abstract Data Type   (ADT) 抽象資料型別
             方法 method == 動作 action == function 函數(函式)
例如:
    Stack 堆疊 ==> 用一個 array, 和一個整數 就可以了

      class Stack{
          double dat_[99];   //寫 class開頭會自動 private:
          int sp;
        public:
          Stack( ) { sp = -1; } // constructor, 要到 Stack位置會自動執行這
          void push(double x) {
             sp++;
             sp = sp % 99;   // 直接寫 99 不好, 怎麼改? 搭配 enum
             dat_[sp] = x;
          }
          double top( ) { return dat_[sp]; }
          void pop( ) { --sp; }
      }; // 別忘了分號
* 這樣 Stack xo, brandy; // 用起來像用物件 --- 堆疊物件

* Inheritance -- class 的繼承就是擴充(extend), 就是請 compiler 幫忙 copy

  ==> 參考講義 mankind 和 Animal 兩個 classes

```
ccbsd3 :cpp % cat -n mannew.h
     1 //mannew.h -- by tsaiwn@csie.nctu.edu.tw
     2 //設計兩個 class: Animal 和 從 Animal 擴充來的 mankind
     3 //Interface file for "mankind" and "Animal" classes
     4 class Animal {
     5         int pv1;
     6         float pv2;
     7     protected:
     8         int pt1[5];
     9     public:
    10         Animal(int = 38);   // default parameter, 若沒寫則compiler自動填38
    11         float pb1;
    12         int pb2[9];
    13         void talk(void);
    14 };
    15 ///
    16 class mankind:public Animal{    // mankind 最好第一字要大寫!
    17         char * pv3;
    18     public:
    19         mankind(char * ="NoName");   // Constructor
    20         ~mankind( );    // Destructor
    21         int pb3[8];
    22         void talk(void);
    23 };
ccbsd3 :cpp % cat -n manlib.cpp
     1 //manlib.cpp -- Implementation file for "Animal" and "mankind"
     2 // 實作 Animal 和 mankind 這兩個 class
     3 // -- by tsaiwn@csie.nctu.edu.tw
     4 #include <stdio.h>
     5 #include <iostream>
     6 using namespace std;
     7 #include "mannew.h"
     8  Animal::Animal(int x){
     9  pv1=x;
    10  pv2=45.67;
    11  this->pb1=135.246;
    12  for (int i=0; i<9; i++) pb2[i]=i+1;
    13  //pb2={ 1,2,3,4,5,6,7 ,8,9};
    14   cout<<" Animal shows up\n";
    15 };
    16 void Animal::talk(void)
    17 {
    18  cout << " Animal talk, pv1=" << dec <<pv1 <<"=0x" << hex << pv1 <<"\n";
    19  cout << " \t pv2=" << pv2 <<"\n";
    20  cout << " \t pb1=" << pb1 <<"\n";
    21  cout << " \t pb2[6]=" << pb2[6] <<"\n";
    22 };
    23 mankind::mankind(char * name){
    24  pv3=name;
    25  cout << " mankind "<< pv3 << " appears\n";
    26 }
    27 mankind::~mankind(void){
    28  cout << " %%% mankind "<< this->pv3 << " is dying \n";
    29 }
    30 void mankind::talk(void)
    31 {
    32   // cout << " mankind talk, pv1=" << pv1 <<"\n";
    33   // cout << " mankind talk, pb2=" << pb2 <<"\n";
    34   // cout << " mankind talk, pb2=" << Animal::pb2 <<"\n";
    35   cout << " mankind talk, pb2[0]=" << pb2[0] <<"\n";
    36  cout << " mankind talk, pv3=" << pv3 <<"\n";
    37  cout << " mankind talk, pb3[3]=" << pb3[3] <<"\n";
    38 };
ccbsd3 :cpp % g++ -c manlib.cpp
ccbsd3 :cpp % ar r libanim.a manlib.o
 ar: creating libanim.a
```

(-c 是說 compile only, 此會生出 manlib.o)

(如果你願意, 可將之做成 library, 則 manlib.cpp 就不用給人家
見下一頁最後; 給 manlib.o 也可以不用給 manlib.cpp; 但 .h 檔要給)

```
ccbsd3 :cpp % cat -n mannew.cpp
     1 //mannew.cpp --test Animal, mankind;  by tsaiwn@csie.nctu.edu.tw
     2 // g++ mannew.cpp manlib.cpp; ./a.out
     3 #include <stdio.h>
     4 #include <iostream>
     5 using namespace std;
     6 #define call
     7 #include "mannew.h"
     8 ///
     9 animal aa1(123),aa2(456);
    10 mankind * mm1, *mm2;
    11 void onlyasub(void)
    12 { mankind nobody;
    13   cout << " Now in routine onlyasub\n";
    14 }
    15 main(){
    16   aa1.talk();
    17   cout << "Welcome to C++\n";
    18   call onlyasub();
    19   mm1= new mankind("Chang-3");
    20   mm2= new mankind("Lee-4");
    21   cout << "mm1->pb1=" << mm1->pb1 << "\n";
    22   cout << " (Let mm1 talk)\n";
    23   mm1->talk();
    24   delete mm2;
    25   cout << " (and then Let mm1 talk by animal method)\n";
    26   mm1->animal::talk();
    27   return(0);
    28 }
ccbsd3 :cpp % g++ mannew.cpp manlib.cpp
ccbsd3 :cpp % g++ mannew.cpp manlib.o
ccbsd3 :cpp % ./a.out | cat -n
     1   Animal shows up
     2   Animal shows up
     3   animal talk, pv1=123=0x7b
     4            pv2=45.67
     5            pb1=135.246
     6            pb2[6]=7
     7  Welcome to C++
     8   Animal shows up
     9   mankind NoName appears
    10   Now in routine onlyasub
    11   %%% mankind NoName is dying
    12   Animal shows up
    13   mankind Chang-3 appears
    14   Animal shows up
    15   mankind Lee-4 appears
    16  mm1->pb1=135.246
    17   (Let mm1 talk)
    18   mankind talk, pb2[0]=1
    19   mankind talk, pv3=Chang-3
    20   mankind talk, pb3[3]=0
    21   %%% mankind Lee-4 is dying
    22   (and then Let mm1 talk by animal method)
    23   animal talk, pv1=38=0x26
    24            pv2=45.67
    25            pb1=135.246
    26            pb2[6]=7
ccbsd3 :cpp % exit


  如何在Unix上造library?
      very simple, 例如:
           g++ -c manlib.cpp     (翻譯出 manlib.o)
           ar r libanim.a manlib.o   (archive 造出 Library  libanim.a )
           ranlib libanim.a    (使它 link 時較快)
        阿這樣用:
                 g++ mannew.C -L. -lanim
                 ( -L. 表示要在 current working directory 找 library,
                   -lanim 表示要找的 library 名爲 libanim.a )
     (For more information, see "man ar"  and "man ranlib")
```