Master M1 SMA Computer Programming Class 2013/14.

Exam 2 hours, open book.

Exercise 1 (2pt)

a) What are the three sets of instruction which control the “for” loop structure (after the for key word in parentheses) ?

b) What is a “scope” ?

c) Where and when the keyword “break” is used ?

d) Does the range of an “int” and the range of an “unsigned int” are the same ? Explain this difference.

Exercise 2 (1pts)

What are “**SVN**”, “**makefiles**” and “**debugging symbols**”.

Exercise 3 (2pt)

a) What is the output when the following code fragment is executed ?

```
1       int n, k = 3;
2       n = (21% k ? k + 1 : k – 1);
3       cout << "n = " << n << " k = " << k << endl;
```

b) What is the output when the following code fragment is executed ?

```
1       int n;
2       int k = 15;
3       int j = 8;
4       k/=3;
5       n = ((k%j)==3)?j/k:k++;
6       cout << "n = " << n << " k = " << k << endl;
```

Exercise 4 (4pt)

In this exercise all array are following c++ norm. The declaration must use a static size.

a) Write a program that ask the user to input one square matrix A of integer, term by term (all of them). Matrix can be of any dimension from 1x1 to 7x7.

The transposed matrix A^t is stored by the program at the same time it read A

Internally those two matrices must be stored this way :

- The first one (A) with a bi-dimensional array.
- The second one (A^t) with a unidimensional array.

b) In a new step the program must multiply all terms of second matrix A^t by 2.

c) In a new step the program must then compute the diagonal terms of internal matrices product (diag(A*(2*A^t))). The result must be stored in a third array.

d) In a last step the program print then this last array containing computational result.

For example this “2x2” matrix :

$$\begin{pmatrix} 1 & 2 \\ 4 & 3 \end{pmatrix}$$

give the following result with this program :

10 50

and this “3x3” matrix

$$\begin{pmatrix} 3 & 2 & 0 \\ 4 & -2 & 1 \\ -1 & 2 & 5 \end{pmatrix}$$

give :

26 42 60

Exercise 5 (3pt)

a) The function addone is supposed to add one to int variable given as argument. Saying if a variable j is equal to 1 then after calling addone(j), this variable must be equal to 2. Does following implementation give the correct behaviour ? Explain why.

```
1      void addone( int  a)
2      {
3           a+=1;
4          return;
5      }
```

b) Write the implementation of the 3 functions declared bellow. They are supposed to swap contain of variable given as argument. Having a variable i and j respectively equal to 1 and 2, after calling any of these functions using i and j, those variables must contain respectively 2 and 1.

```
1      void swap1( int & x1, int & x2);
2      void swap2( int & x1, int * x2);
3      void swap3( int * x1, int * x2);
```

c) Write a main function which call these 3 functions and illustrate their use.

Exercise 6 (2pt)

What is the output of the following program ?

```
1      #include <iostream>
2      using namespace std;
3      int main()
4      {
5           int a[] = {24,5,90,2,60,23};
6           int *p=a+2;
7           int *q=&a[4];
8           cout <<  (*p)+1 << " " << *(q+1) << endl;
9           cout <<  *(++p) << " " << *((q--)-3) << endl;
10          cout << (p[0])++ << " " << q[1] << endl;
11          cout <<a[0]<<" "<<a[1]<<" "<<a[2]<<" "<<a[3]<<" "<<a[4]<<"
       "<<a[5]<<endl;
12          return 0;
13     }
```

Exercise 7 (4pt)

Here is the declaration of a class hierarchy stored in mesh.h below :

```
1       class AttachableData
2       {
3         public :
4            void attacheData(int i, double v);
5            double & getData(int i);
6         private :
7           double data[2];
8       };
9       class Entity : public AttachableData
10      {
11        public :
12           Entity(int id);
13           int getId(void);
14           virtual void print(void);
15        private :
16           int ID;
17      };
18      class Vertex : public  Entity
19      {
20        public :
21           Vertex (int id, double const &x, double const &y, double const &z);
22           void print(void);
23        private :
24           double  coord[3];
25      };
26      class Face : public  Entity
27      {
28        public :
29           Face (int id, Vertex *v1, Vertex *v2, Vertex *v3);
30           void print(void);
31        private :
32           Vertex *connect[3];
33      };
```

It is extracted from a mesh database library. This library offer capability to store information attached to any kind of entity of the mesh. Here in this extra-light version only 2 double information may be attached to vertex or faces.

Here is a main program, stored in main.cc, which use this class hierarchy :

```
1       #include "mesh.h"
2       int main ()
3       {
4         Entity *c[6];
5         Vertex n1(10,0.,0.,0.);
6         c[0]=&n1;
7         Vertex n2(20,0.,1.,0.);
8         c[1]=&n2;
9         Vertex n3(30,1.,0.,0.);
10        c[2]=&n3;
11        Vertex n4(40,1.,1.,0.);
12        c[3]=&n4;
13        Face f1(1,&n1,&n3,&n2);
14        c[4]=&f1;
15        Face f2(2,&n3,&n4,&n2);
16        c[5]=&f2;
17        int data0=0;
18        int data1=1;
19        for (int i=0;i<6;i++)
20        {
21          (c[i])->attacheData(data0,i*2.);
22          (c[i])->attacheData(data1,6.-i);
23          (c[i])->print();
24        }
25        return 0;
26      }
```

The expected output of this executable is :

```
1       id is 10 attached data 0 is 0 attached data 1 is 6
2       coordinates are : 0 0 0
3       id is 20 attached data 0 is 2 attached data 1 is 5
4       coordinates are : 0 1 0
5       id is 30 attached data 0 is 4 attached data 1 is 4
6       coordinates are : 1 0 0
7       id is 40 attached data 0 is 6 attached data 1 is 3
8       coordinates are : 1 1 0
9       id is 1 attached data 0 is 8 attached data 1 is 2
10      conectivity is : 10 30 20
11      id is 2 attached data 0 is 10 attached data 1 is 1
12      conectivity is : 30 40 20
```

a) In main.cc why does pointer to n1,n2 ..., f2 may be stored in c array ?

b) In which class is it possible to access “data array” member (mesh.h line 7) ? Explain.

c) In which class is it possible to access “ID” member (mesh.h line 16) ? Explain.

d) Write the implementation of this hierarchy in a mesh.cc file so that you get expected output above with given main above. The mesh.h and main.cc are supposed to be unchanged. 9 methods should be implemented (counting constructor). Obviously Entity constructor set ID member data. Constructor for Vertex set coord and ID. Constructor for Face set connect and ID. “print” function display ID, data, coord and/or connect information stored in instance which call them.

Exercise 8 (2 pts)

a) What is a function template ?

Here is a simple function template :

```
1       template <typename T, typename U>
2       U compute (const T &a, const T &b)
3       {
4           T tmp = a/b;
5           U one(1.);
6           U res=one+tmp;
7           return res;
8       };
```

b) What are the requirements on the template parameter T for the function compute to be use for a given type ?

c) What are the requirements on the template parameter U for the function compute to be use for a given type ?

d) Write a main which call compute function with :

- Arguments are two variable x (a) and y (b) of type 'float' respectively equal to 3 and 4.
- returned value of the function must be stored in an int variable i. Type of the function returned value must be the same as this int i variable.

e) What is the value of i after the call.