Master M1 SMA Computer Programming Class 2014/15.

Exam 2 hours, open book.

Exercise 1 (2pt)

a) How is a function declared ? Give all components of this declaration.

b) Why a variable created in an "if" structure block is not available after this block ?

Having :

```
1       int *p1=NULL;
2       if ( "some condition" )
3       {
4           p1 = new int [4];
5           int *p2 = new int [4];
6       }
7       "some instruction"
```

Is p2 available line 7 ?

Is p1 available line 7 ?

Is the dynamic memory allocated line 4 exist when executing line 7 ?

Is the dynamic memory allocated line 5 exist when executing line 7 ?

c) Which loop structure ensure that, independently of test condition, the set of instructions related to this loop are always at least executed once ?

Exercise 2 (1pts)

a) When is it useful to use a Version Control System ?

b) How "**make/makefiles**" helps programmers ?

Exercise 3 (1pt)

What are the different mechanisms available in C++ to pass information contained in variables to a function (not function template).

Exercise 4 (3pt)

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

```
1           int n,k;
2           int j = 8;
3           for (int i = 0; i < 3; ++i)
4           {
5               switch (i)
6               {
7                   case 0 :
8                       k = 2;
9                       n = (( ( k )%( j-- ) ) == 1 ) ? j/k : k++;
10                      break;
11                  case 1 :
12                      j--;
13                      k += 3;
14                      n = ( k%j ) ? ( ++k ) : n+2;
15                  case 2 :
16                      k = ( k == n || k == j ) ? ( n-k ) : n+k;
17                      n *= ( i+1 )-k/2;
18                      break;
19              }
20              cout << "n = " << n << " k = " << k << endl;
21          }
```

Exercise 5 (1,5pt)

What is the output of the following program ?

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

Exercise 6 (4pt)

a) We have to implement a class hierarchy in a library. Base class “A” must have one public pure virtual method “f1” and one none virtual method “f2”. Class “B” and “C” derive publicly from A and have 2 public non virtual methods “f1” and “f2” each. Implementation of “f1”,”f2” in B and C is just the output with “cout” of a message saying “in function fX of class Y” where X and Y stand for 1 or 2 and B or C respectively. “f2” of class “A” output “in function f2 of class A”. “f1”,”f2” are methods

taking no argument and returning nothing (void).
Write this hierarchy and it's implementation. You must separate declaration from implementation by putting on your sheet of paper the starting and ending of files that you would have used to place this code in, if you really use a computer. If typed in files, your solution must compile (i.e. put every details).

b) Using your library, a main program include the following lines. Which line(s) is(are) false if any :

```
1       A *pa;
2       A *pb;
3       A *pc;
4       A a;
5       B b;
6       C c;
7       pa = &a;
8       pb = &b;
9       pc = &c;
10      pa->f1();
11      pa->f2();
12      pb->f1();
13      pb->f2();
14      pc->f1();
15      pc->f2();
```

c) Assuming that the eventually incorrect instructions in the code above had been removed, what is the output of each correct instruction when the program is run ?

d) Imagine now that someone introduce in the implementation part of your library, an implementation of class “A” “f1” method, outputting "in function f1 of class A".

Why is it possible ? Does it change anything to previous “b” and “c” question ?

e) Inside “f1” method of class “B” we want to call “f1” method of class “A”, what syntax should you use ?

Exercise 7 (2 pts)

a) In a class “A” how do you write the declaration of a method to overload the “*” arithmetic operation (multiplication of 2 instances of class “A”).

b) Imagine now that “A” represent special complex number that deal with real and imaginary part in rational arithmetic. Private members are integer “a”,”b”,”c” and “d”. In mathematical notation an instance of this class is represented by following formula :

$\frac{a}{b} + i.\frac{c}{d}$ where as usual $i^2 = -1$

Write the implementation of the method of question “a”. The following code gives an example of the usage of class “A”. Note that the constructor is obviously setting private member using arguments (something like A(int a_, int b_, int c_, int d_) :a(a_),b(b_),c(c_),d(d_){}) and that function exist to output instance of “A” (the same way as it is presented in comment of lines 1 and 2).

```
1       A c1(1,2,3,2); // 1/2 + i.3/2
2       A c2(1,3,2,3); // 1/3 + i.2/3
3       A c3=c1*c2;
4
5       cout<<c3<<endl;
```

If run this piece of code should gives the following result :

$-30/36 + i.30/36$ or $-5/6 + i.5/6$ depending on how you implement your "*" method.

Exercise 8 (2,5 pts)

a) Here is a simple function template declaration :

```
1       template <typename T>
2       double f1 ( T a, int i);
```

In this example what is the difference between parameter T and function argument ("a" and "i") ? What kind of information do they provide to function "f1" ?

b) Here is an implementation of this function :

```
1       template <typename T>
2       double f1 ( T a, int i)
3       {
4           double res=a[i];
5
6           return sgrt(res);
7       };
```

What are the requirements of the generic type T used in function "f1" ?

c) Write a piece of program which call "f1" function with :

- "a" argument being an instance of a simple class fulfilling requirement of "f1" function. You must give the declaration and implementation of the class you used. All the required method for function "f1" must appear and be implemented and must do something (no empty method(s)).
- "a" arguments being a variable of a standard kind of the language (not an instance of a new class or struct like above).

Exercise 9 (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 symmetric matrix A of integer, term by term. Matrix can be of any dimension (called n hereafter) from 1x1 to 8x8. Only the upper triangle part including diagonal is asked and stored .

Internally this matrix must be stored in a uni dimensional array.

Refering to the example matrices below, this program only store 3 terms (1 2 3) for the 2x2 matrix and 6 terms (3 2 0 -2 1 5) for the 3x3 matrix. For a matrix of dimension n, n*(n+1)/2 terms are stored.

b) In a new step the program must ask the user to enter a vector V of dimension n.

Internally this vector must be stored in a uni dimensional array.

c) In a new step the program must compute the matrix vector product A*V. The result must be stored in a third vector array R of dimension n.

R component are given by following general mathematical formula : $R_i = \sum_{j=1}^{n} A_{ij}.V_j \quad \forall i \in [1,n]$

You must implement this product using this formula has general guidance and taking into account the fact that you have only in memory A_{ij} *terms with* $j \geq i \quad i \in [1,n] \quad j \in [1,n]$

d) In a last step the program must print the R array.

For example this “2x2” matrix :

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

with this vector :

$$\begin{pmatrix} 4 \\ 5 \end{pmatrix}$$

give the following result with this program :

14 23

and this “3x3” matrix

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

with this vector :

$$\begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix}$$

give :

22 4 35