Exam 2 hours, open book.

Exercise 1 (1,5pt)

C++ is a typed language. This mean that all variable of a program have a “type”.

a) How does the "type" of a variable is chosen ?

b) "type" is also important when talking about pointer. Give 2 operators (of different nature) related to pointer that use “type”. Explain how those types are used with those operators.

c) In object oriented programming what is defining a “type“ ?

Exercise 2 (1pts)

a) What can be inspected with a debugger ? Does a class private member of an instance can be inspected with a debugger ?

b) When and where does the documentation effort take place when using Doxygen ?

Exercise 3 (1pt)

What is the signature of a function ? What are the elements involved in this signature (4 expected)?

Exercise 4 (3pt)

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

```
1           int n = 2, k = 3;
2           int i = 2, j = 7;
3           do
4           {
5               switch (i)
6               {
7                   case 0 :
8                        k *= 4;
9                        n = (( ( k )%( ++j ) ) == 1 ) ? j/k : k-31;
10                  case 1 :
11                      j--;
12                      k += j;
13                      n += ( k%j ) ? ( ++k ) : n+2;
14                      i = k%2;
15                      break;
16                  case 2 :
17                      k = ( i == n && k == j ) ? ( n+k ) : k-n;
18                      n += ( i+1 )-k/2;
19                      i = k%3;
20                      break;
21              }
22              cout << "n = " << n << " k = " << k << endl;
23          } while(k<22);
```

Exercise 5 (3pt)

What is the output of the following program ?

```
1       #include <iostream>
2       using namespace std;
3       void f1(int a, int *p)
4       {
5            (*p)+=a;
6       }
7       void f2(int a, int *p)
8       {
9            a+=(*p);
10      }
11      void f3(int *p)
12      {
13           ++p;
14      }
15      void f4(int *p)
16      {
17          int tmp,*p0 = p;
18          int *p3 = p+3;
19          while (p0 < p3)
20          {
21              tmp = *p3;
22              *p3-- = *p0;
23              *p0++ = tmp;
24          }
```

```
25     }
26     int main()
27     {
28          int a[] = {3,9,4,6};
29          int *p=a+2;
30          int *r=&a[1];
31          cout <<  (*p)-2 << " " << *(r+1) << endl;
32          cout <<  (*(r++))++ << " ";
33          cout << ((p==(r--))?(a[0])++:(a[3])++ ) << endl;
34          cout << ++(r[1]) << " " << (p-r) << endl;
35          f1(a[0],p);
36          f2(a[1],p);
37          f3(r);
38          f4(a);
39          cout <<a[0]<<" "<<a[1]<<" "<<a[2]<<" "<<a[3]<<endl;
40          return 0;
41     }
```

Exercise 6 (3,5pt)

a) We have to implement a class hierarchy in a library. Base class “A” must have one public virtual method “f1” and one none virtual method “f2”. Class “B” derive publicly from A and have one public pure virtual method “f1” and one none virtual method “f2”. Class “C” derive publicly from B and have two public none virtual method “f1” and “f2”.

Implementation of “f1”and ”f2” method in class A, B and C is just the output with “cout” of a message saying “in fX of class Y” where X and Y stand for 1 or 2 and A, B or C respectively, except for pure virtual method “f1” of class B that won't be implement.

For example “f2” method of class “A” output “in 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 a;
2      B b;
3      C c;
4      A *p1 = &a;
5      p1->f1();
6      p1->f2();
7      A *p2 = &b;
8      p2->f1();
9      p2->f2();
10     A *p3 = &c;
11     p3->f1();
12     p3->f2();
13     B *p4 = &b;
14     p4->f1();
15     p4->f2();
16     B *p5 = &c;
```

```
17      p5->f1();
18      p5->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 “f2” method of class “B” is private. Does it change anything to previous “b” and “c” question and if yes what are the correction if any and the new program output?

e) Imagine now that “f2” method of class “B” is virtual (and public). Does it change anything to previous “b” and “c” question and if yes what are the correction if any and the new program output?

Exercise 7 (3,5 pts)

Imagine a class “A” that represent special complex number that deal with real and imaginary part of huge or tiny real number treated like in class “Real” of Labs. Private members are double “mr”, ”mi” and integer “er” and ”ei”. In mathematical notation an instance of this class is represented by following formula :

$$mr.10^{er} + i.mi.10^{ei}$$

where as usual $i^2=-1$ and like in labs $mr\in]-1,-0.1]\cup[0.1,1[$ and $mi\in]-1,-0.1]\cup[0.1,1[$

You will consider that all declaration and implementation are done somehow somewhere. In particular a constructor exist and is setting private member using its arguments in the following way :

```
1       A::A(double mr_, int er_, double mi_, int ei_) :
2       mr(mr_),mi(mi_),er(er_),ei(ei_)
3       {
4       /* like in labs, there is something here that modify mantissa
5        * mr and mi (and er and ei) to be in correct interval.
6        * Implementation is not given here but it exists.
7        */
8       …
9       }
```

There exist also a function to output instance of “A” in the same way as it is presented in comment of lines 1 to 4 of the following code which gives an example of class “A” usage.

```
1       A c1(0.2,200,0.5,300); // 0.2E200 + i0.5E300
2       A c2(10.,400,3.,300); // 0.1E402 + i0.3E301
3       A c3(1.,1,2.,2); // 0.1E2 + i0.2E3
4       A c4(3.,2,4.,1); // 0.3E3 + i0.4E2
5       A c5=c1*c2;
6       A c6=c3*c4;
7
8       cout<<c5<<endl;
9       cout<<c6<<endl;
```

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

0.5E600 + i0.5E701

-0.5E4 + i0.604E5

a) Implement a method to overload the "+" arithmetic operation (addition of 2 instances of class "A"). Like in labs you will consider that if the difference of exponent (for real or imaginary part) is greater than 20 then the number with the smallest exponent is negligible compared to the number with the biggest exponent and will be considered null for addition. This test have to be done independently for real and imaginary part. Otherwise, the operations will rely on " double " floating-point precision. Write only implementation of this operator on your copy just like if it was written in a .cc containing class "A" implementation.

b) Implement a method to overload the "*" arithmetic operation (multiplication of 2 instances of class "A"). You **must** use overloaded "+" arithmetic operation of a) in your implementation. Write only implementation of this operator on your copy just like if it was written in a .cc containing class "A" implementation.

Exercise 8 (3,5 pts)

a) Here is a function template declaration :

```
1       template <typename T, typename U>
2       double f1 ( T a, U & f);
```

Here is an implementation of this function :

```
1       template <typename T, typename U>
2       double f1 ( T a, U & f)
3       {
4           T two(2);
5           double res=f(two*a);
6
7           return res*res;
8       };
```

Imagine that those lines (declaration + implementation) are written in a "f1.h" (or declaration in "f1.h" which include "f1_imp.h" containing implementation).

How do you compile a main.cc including "f1.h" containing a main entry point function which use f1 ? Write the precise sequence of command(s) you would have used on a terminal to generate an executable.

b)What are the requirements imposed to the generic type T by function "f1" ?

c)What are the requirements imposed to the generic type U by function "f1" ?

d) Write a main.cc files mentioned above to illustrate how f1 can be used (i.e. call at least 2 time f1 in your main function using different types that you may have to invent).

e) Propose for "f1" another interface (your main from question d) becomes obsolete) that only rely on one template parameter. The implementation of "f1" function must stay exactly the same. The idea is that arguments given to "f1" are now compatible and this is checked at compilation time.