

# DM560 — Introduction to Programming in C++

## Class Test 1, Autumn 2018

---

YOUR NAME:

USERID:

---

**Solution:** Included.

Each exercise gives 5 points. There are 100 points to collect overall.

### Exercise 1

For each of the following three lines, write what is the error and its type (compiler error, run time error, logic error).

```
vector<char> v(5); for (int i=0; i<=v.size(); ++i) cout << v[i] << endl; cout << "Success!\n " ;
```

**Solution:** No error, it was meant to be:

```
vector<char> v(5); for (int i=0; i<v.size(); ++i) cout << v[i] << '\n " ;
```

which would have give: run time error: index runs over the limit of the vector size.

```
int i=0; int j = 9; while (i<10) ++j; if (j<i) cout << " Success!\n " ;
```

**Solution:** logic error: the loop never terminates, the index is not incremented.

```
cin << "Success!\n " ;
```

**Solution:** compiler error: no match for operator<< with const

### Exercise 2

Determine the type of *g* in this piece of code:

```
int a = 3;  
double d = 2.19;  
auto g = (d + 2)/a;
```

**Solution:** double

### Exercise 3

Find the value stored in each variable at the end of the execution:

```
double b = 3.1, c = 0.0;
c = c + 2.0;
b = b * (c + 3.0);
int i = (int) (c + b);
i = i - 1;
```

b = 15.5  
c = 2  
i = 16

### Exercise 4

Your program consists of the following files:

```
Point.h
struct Point {...};
...

Window.h
class Window {...};
...

Window.cpp
#include "Window.h"
...

Simple_Window.h
#include "Window.h"
...

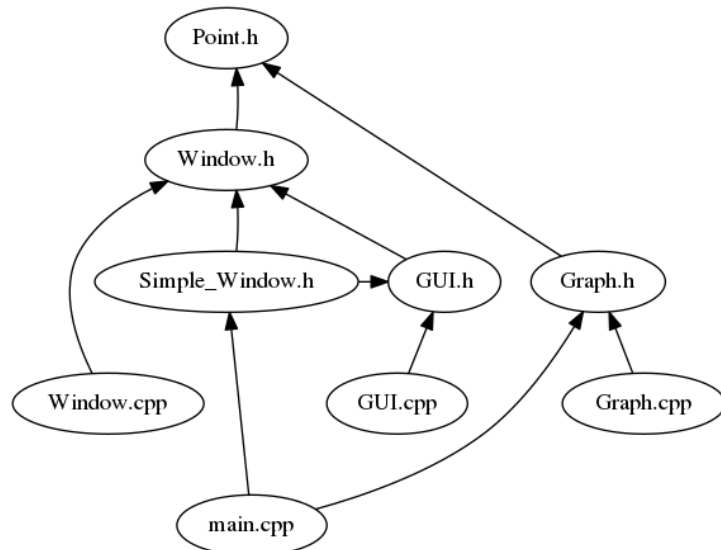
GUI.h
#include "Window.h"
struct In_box {...};
...

GUI.cpp
#include "GUI.h"
...

Graph.h
#include "Point.h"
...

Graph.cpp
#include "Graph.h"
...

main.cpp
#include "Graph.h"
#include "Simple_Window.h"
```



Write the commands that you would write in a shell of the Computer Lab to build the program from these files.

**Solution:**

```
g++ -c Window.cpp GUI.cpp Graph.cpp main.cpp
g++ -o myprogram Window.o GUI.o Graph.o main.o
```

This would also work:

```
g++ -c *.cpp
g++ -o myprogram *.o
```

and also this:

```
g++ *.cpp
```

**Exercise 5**

Write all errors that compiling this file will produce

```
// main.cpp

int main()
{
    cout << f(i) << '\n';
}
```

**Solution:** cout is not defined, f is not defined, i is not defined

**Exercise 6**

Is this a definition or a declaration?

```
double calculate_fraction(double);
```

**Solution:** A declaration

**Exercise 7**

Suppose we need to work with the following data:

- a distance between stars, measured in light-years;
- a distance on the Earth's surface, measured in meters.

Propose names and types for variables to store these data.

**Solution:** Here many solutions are possible but the names must be self explanatory and be in correct syntax.

```
distance_stars_in_light_years  
earth_distance_in_meters
```

Maximum length of a variable name:

- Microsoft C++: 2048 characters
- Intel C++: 2048 characters
- g++: No limit, as answered in this question: is there a length limit on g++ variable names?

## Exercise 8

Write a function `void join(const vector<int> &v, const vector<int> &w)` that outputs the elements of `v` followed by the elements of `w` (in the original order).

**Solution:**

```
void join(const vector<int> &v, const vector<int> &w)  
{  
    for (int a : v)  
    {  
        cout << a << endl;  
    }  
    for (int a : w)  
    {  
        cout << a << endl;  
    }  
}
```

## Exercise 9

Write a method `int factorial(int n)` that returns the factorial of `n` for  $n \geq 0$ .

**Solution:**

```
#include <iostream>  
  
using namespace std;  
  
int factorial(int n) {  
    if (n<=1) return 1;  
    return n*factorial(n-1);  
}  
  
int main() {  
    cout << factorial(4);  
}
```

## Exercise 10

Write a method `int occur(const vector<string>& s, const string& word)` that returns the number of occurrences of the word `word` in `s`.

**Solution:**

```
int factorial(const vector<string>& s, const string& word) {
    int count=0;
    for (string ss : s) {
        if (ss == word)
            count++;
    }
    return count;
}
```

## Exercise 11

Correct the following piece of code by defining the two functions `do_smth` inside two different namespaces, `N1` and `N2`, and then ensuring that the function used in the main function is the one defined in namespace `N1`.

```
#include<iostream>

void do_smth() {
    cout<<','<<'S'<<'mth from N1'<<endl;
}

void do_smth() {
    cout<<','<<'S'<<'mth from N2'<<endl;
}

int main() {
    do_smth();
    return(0);
}
```

**Solution:**

```
#include<iostream>

N1 {
    void do_smth() {
        cout<<','<<'S'<<'mth from N1'<<endl;
    }
}

N2 {
    void do_smth() {
        cout<<','<<'S'<<'mth from N2'<<endl;
    }
}

int main() {
    N1::do_smth();
    return(0);
}
```

## Exercise 12

What is the outcome of the following piece of code:

```
#include<iostream>
using namespace std;

int x;
int main()
{
    cout << x << endl;
}
```

Unpredictable, Runtime error, or 0?

**Solution:**

0 or Unpredictable

## Exercise 13

Predict the output of following C++ program?

```
#include<iostream>
using namespace std;

int main()
{
    int x = 10;
    int& ref = x;
    ref = 20;
    cout << "x = " << x << ' '; ' ';
    x = 30;
    cout << "ref = " << ref << endl;
    return 0;
}
```

x = 20  
ref = 30

## Exercise 14

What is the output of the following piece of code?

```
#include <iostream>
using namespace std;
int main()
{
    int x = -1;
    try {
        cout << "Inside try\n";
        if (x < 0)
        {
            throw x;
            cout << "After throw\n";
        }
    }
    catch (int x ) {
        cout << "Exception Caught\n";
    }
    catch (...) {
        cout << "Default Exception Caught\n";
    }
}
```

```
}

cout << "After catch\n";
return 0;
}
```

**Solution:**

```
Inside try
Exception Caught
After catch
```

**Exercise 15**

What is the output of the following piece of code?

```
#include <iostream>
using namespace std;

int main()
{
    try
    {
        try
        {
            throw 20;
        }
        catch (int n)
        {
            cout << "Inner Catch\n";
            throw;
        }
    }
    catch (int x)
    {
        cout << "Outer Catch\n";
    }
    return 0;
}
```

**Solution:**

```
Inner Catch
Outer Catch
```

The throw without argument in the inner loop throws the same exception that was caught. This was written in the book and not in the slides, a answer with only the first line would also be accepted as correct.

**Exercise 16**

What happens in C++ when an exception is thrown and not caught anywhere like following program.

```
#include <iostream>
using namespace std;

int fun() throw (int)
{
```

```
        throw 10;
    }

    int main() {
        fun();
        return 0;
    }
```

- A Compiler error
- B Abnormal program termination
- C Program doesn't print anything and terminates normally

**Solution: B**

## Exercise 17

Predict the output of the following program:

```
#include <iostream>
using namespace std;

int main()
{
    const int x;
    x = 10;
    cout << x << endl;
    return 0;
}
```

- A. Compiler Error
- B. 10
- C. 0
- D. Runtime Error

**Solution: A.** A variable declared const must be initialized together with its definition and it cannot be changed afterwards.

## Exercise 18

What is the output of the following C++ program?

```
#include <iostream>
int const s=9;
int main()
{
    std::cout << s;
    return 0;
}
```

- A. Compiler Error
- B. 9
- C. 0



**Solution: B**

## Exercise 19

What is the output of the following C++ program?

```
#include <iostream>
using namespace std;

int fun(int x = 0, int y = 0, int z)
{ return (x + y + z); }

int main()
{
    cout << fun(10);
    return 0;
}
```

- A. 10
- B. 0
- C. 20
- D. Compile error

**Solution: D: parameters with default values must be written at the end of the function**

## Exercise 20

What is the output of the following program?

```
#include <iostream>
using namespace std;
int f(int n)
{
    cout << n << endl;
    if (n <= 1)
        return n;
    return f(n-1) + f(n-2);
}

int main ()
{
    int n = 3;
    cout << f(n);
    return 0;
}
```

**Solution: 3 2 1 0 1 2**

The function calculates in a recursive way the Fibonacci numbers. The method is inefficient, it runs in exponential time, while a polynomial time algorithm exists.