Linköping University
Department of Science and Technology/ITN
Aida Nordman

Introduction to Programming (C++)

TND012

Laboration 2

Course goals

- To write programs with selection statements: **`if`**-statement and **`switch`**-statement.

Preparation

You must perform the tasks listed below before the start of the lab session in week 36. In each lab session, your lab assistant has the possibility to discuss about three questions with each group. To make the best use of your time during the lab session, it is important that you read the description of all exercises in this lab and do the indicated preparation steps.

- Review the concepts and examples presented in Fö 2.
- Review lesson 1 exercises.
- Read sections 5.1 and 5.2 of course book.
- Download the source file **`uppgift1.cpp`**.
- Do exercise 1 and exercise 2.
- Do steps a. and b. of exercise 3.

Your source files should be placed in the folder **`TND012\Labs\Lab2`**.

Presenting solutions and deadline

The three exercises in this lab are compulsory and you should demonstrate your solutions during your lab session in **week 36**. After week 36, if your solution for lab 2 has not been approved then it is considered a late lab. This lab must be presented latest in your lab session in week 38. We also remind you that your code for the lab exercises cannot be sent by email to the staff.

We remind you that in this course a late lab can be presented in another lab session provided there is time.

Before presenting your code make sure that it is readable, well-indented, and that the compiler issues no warnings. Otherwise, your lab won't be approved.

If you have any specific question about the exercises, then send us an email. Be short and concise, otherwise you won't get a quick answer. You can write your email in Swedish. Add the course code and your study programme to the e-mail's subject, e.g. "**TND012/ED:** …"..

Exercise 1

Download the source file **uppgift1.cpp** and try to understand the program. Make sure the compiler settings listed in the appendix of lab 1 are set on. Then answer the questions below.

- What output would you expect from the program?
- What is the output obtained when you run the program?
- If the expected output differs from the actual output produced by the program, then investigate a possible explanation for this behaviour. What conclusions did you reach?
- Check whether the compiler produced any warning message after compilation and write it down.

In the beginning of this lab session, your lab assistant will discuss this exercise with you.

Exercise 2

Create a new program file (source file) named **uppgift2.cpp**. Write a program that reads in a value (**double**) and assigns it to an **int**. In a second step, read in an **int** and assign it to a **double**. Print both assigned values.

- What output would you expect from the program?
- Do you see any problem with the code? If so, write it down.
- Check whether the compiler produced any warning message after compilation and write it down.

In the second part of this exercise, you are supposed to investigate the number of bits for various data types and their respective largest and smallest values. To identify the number of bits, the **sizeof()** function can be used, e.g. **sizeof(int)**.

- Extend **uppgift2.cpp** and show the size in bytes for the following types **int**, **long int**, **long long int**, **float**, and **double**.

  Note that variables `i1`, `i2`, and `i3`, declared below, should store integer values, while variables `x` and `y` should store real values.

```
int i1;
long int i2;
long long int i3;
```

```
float x;
double y;
```

- To identify the smallest and largest values, e.g. for an **int**, you should use **numeric_limits<int>::lowest()** and **numeric_limits<int>::max()**. Do not forget to include **<limits>**. Do the same for all the other types above.
- Compare the results with your colleagues. Is there a difference?

Exercise 3

Write a program that displays the price of a football match ticket. Ticket prices depend on the age of the ticket's owner.

- If the owner is over 15 years old then the ticket costs 80 SEK.
- Otherwise, if the owner is at least 8 years old then he pays 30 SEK.
- Children younger then 8 years can get a ticket for free.

Some running examples of the program are given below (user input shown in green).

```
Welcome to our Football Arena.
------------------------------
Enter age: 12
Price = 30 SEK

Welcome to our Football Arena.
------------------------------
Enter age: 20
Price = 80 SEK
```

To solve this exercise, proceed as indicated below.

a. Create a new program file (source file) named **`uppgift3.cpp`**.

b. Write an algorithm for this problem which indicates the major steps of the program. Note that these steps should be written in Swedish or English, within comments, in the **`main`** function.

c. Encode each of the steps in your algorithm in C++ (**`if`**-statements should be used), **one step at a time**. Then compile and run the program before you proceed to the next step of the algorithm.

   We remind you that it is important that you correct any compilation errors and test if the program is behaving as expected after encoding each step. Only then you should proceed to the next step of your algorithm.

d. Avoid the use of magic constants (e.g. **`80`**, **`15`**) in the program. Instead, define constants with suitable names.

Exercise 4

Write a program that computes and displays the real roots of a second degree equation of the form $x^2 + bx + c = 0$, with b and c real constants, if there are real roots. For instance,

- $x^2 + 3x - 4 = 0$ is a second degree equation with roots $x_1 = 1$ and $x_2 = -4$;
- $x^2 - 5x - 12.5 = 0$ is another second degree equation with roots $x_1 = 6.83$ and $x_2 = -1.83$.

The roots of a second degree equation as above can be obtained through the well-known formula

$$x_1, x_2 = -\frac{b}{2} \pm \sqrt{\left(\frac{b}{2}\right)^2 - c}\,.$$

Note that for some second degree equations both roots x_1 and x_2 are the same, i.e. there is a double root. For other second degree equations, it may not exist real roots, but only complex roots.

- **Question:** When are there no real roots of a second degree equation?

Your program should start by asking the user the values for constants b and c. Then, it should write the roots of the equation with three decimal digits of precision, if real roots exist. Otherwise, the program should inform the user that only complex roots exist.

Below, you can find some examples when running the program (user input shown in green).

To solve this exercise, proceed as indicated below.

a. Create a new program file (source file) named **`uppgift4.cpp`**.

b. Write an algorithm for this problem that indicates the major steps of the program. Note that these steps should be written in Swedish or English, within comments, in the **`main`** function.

c. Encode in C++ each of the steps in your algorithm, **one step at a time**. Then, compile and run the program before you proceed to the next step in the algorithm.

d. Make sure the compiler generates no warning messages.

Example 1

```
Solving a second degree equation
of the form x^2 + b*x + c

Enter coefficient b: 3
Enter coefficient c: -4

x1 = 1.000
x2 = -4.000
```

Example 2

```
Solving a second degree equation
of the form x^2 + b*x + c

Enter coefficient b: -5
Enter coefficient c: -12.5

x1 = 6.830
x2 = -1.830
```

Example 3

```
Solving a second degree equation
of the form x^2 + b*x + c

Enter coefficient b: 1
Enter coefficient c: 0.25

x1, x2 = -0.500   (double root)
```

Example 4

```
Solving a second degree equation
of the form x^2 + b*x + c

Enter coefficient b: 2
Enter coefficient c: 4

x1 and x2 are complex roots !!
```

Exercise 5

Write a program that computes the arrival time of a flight. The input data of your program should be

- the departure time of the flight, hours and minutes, and
- the flight duration, hours and minutes. Assume that no flight duration is over $24h$.

Then, your program should output the arrival time of the flight in the form `HH:MM` (e.g. 05:20). Your program should also validate the user input, e.g. minutes should be larger or equal than 0 but smaller than 60. If the user input is invalid an error message is displayed and the program terminates.

Note that a flight may depart at one day and arrive the day after.

To solve this exercise, proceed as indicated below.

a. Create a new source file named **`uppgift5.cpp`**.

b. Write an algorithm for this problem that indicates the major steps of the program. Note that these steps should be written in Swedish or English, within comments, in the **`main`** function.

c. Encode in C++ each of the steps in your algorithm, **one step at a time**. Then, compile and run the program before you proceed to the next step in the algorithm.

Lycka till ☺