
EEE2007: Computer Systems and Microprocessors

Lab 4: Review of C/C++ (Re-visiting input/outputs and basic functions)

Module Instructor: Dr Rishad Shafik

Notes and Examples available in: <http://www.rishadshafik.net/teaching.html>
under **** Lab Notes and Examples ****

Exercise I: Review of C/C++ inputs and outputs

Recommended Time: 20 Mins Maximum

Aim: To understand how inputs and outputs are taken for different variable types

Follow the instructions below and try to do accordingly-

1. **DOWNLOAD** the source code of Lab4ex1.cpp from [HERE](#).
2. **REVIEW** the source code of Lab4ex1.cpp

Go through each line to understand how the code is organized. Check the following:

- A character variable is input from console and also printed at the output console
- A integer variable is input from console and also printed at the output console
- A float variable is input from console and also printed at the output console
- A *cin.ignore()* is used to flush the input buffer as the next input will require cleared buffer
- A character array is input from console and also printed at the output console
- How **cin** and **cout** are used to input and output different variables
- How namespace is used to avoid specifying **std::cin** and **std::cout**.

3. **COMPILE** the source code of Lab4ex1.cpp:
 - a. Start Cygwin command shell through Start->All Programs->Cygwin->Cygwin Bash Shell
 - b. In the Cygwin shell type: `g++ -Wall Lab4ex1.cpp -o Lab4ex1`
The `-Wall` option enables all the warnings.
The `-o` option enables specification of the output executable

Your compilation should generate an executable called Lab4ex1.exe

4. **EXECUTE** the executable by typing the following in the Cygwin shell
`./Lab4ex1.exe`
5. **EXERCISE:**
 - a. Try entering characters when taking integer inputs, or float inputs, see what happens.
 - b. What is the function of return o?
 - c. Research further why *cin.ignore()* is used.

Exercise II: Review of C/C++ basic functions

Recommended Time: 25 Mins Maximum

Aims:

- a. To learn to create functions,
- b. To understand how functions create different variable scopes, and
- c. To understand how functions make a program more modular.

Follow the instructions below and try to do accordingly-

6. **DOWNLOAD** the source code of Lab4ex2.cpp from [HERE](#).

7. **REVIEW** the source code of Lab4ex2.cpp

Go through each line to understand how the code is organized. Check the following:

- A function is used to take input from the standard console
- The main function has one character variable: var_char, while the function has its own local variable: local_var_input
- The return from the function creates a channel for passing the character input from function char_input_function to main function.
- Notice how the char_input_function() is used in an assignment expression in the main function.

8. **COMPILE** the source code of Lab4ex2.cpp as follows:

- a. Start Cygwin command shell through Start->All Programs->Cygwin->Cygwin Bash Shell
- b. In the Cygwin shell type: `g++ -Wall Lab4ex2.cpp -o Lab4ex2`

The `-Wall` option enables all the warnings.

The `-o` option enables specification of the output executable

Your compilation should generate an executable called Lab4ex2.exe

9. **EXECUTE** the executable by

`./Lab4ex2.exe`

10. **EXERCISE:**

- a. Create functions for taking input for each other variable type: int, float, and char array.
- b. *Hint:* for character array input your return type will need a "char *". "Char *" is a way of returning character arrays with unspecified size.

Exercise III: Review of C inputs, outputs and functions (and using arrays)

Recommended Tim: 60 Mins Maximum

11. **WRITE** a C/C++ program that can take 5 different numbers and return the average value. Your program will include:
 - a. A function that can take an integer input.
 - b. Your program will call this function 5 times for five different variables.
 - c. Another function that will take five integer values and return the average value (in floating type).
 - d. Your main function will call this function once all 5 variables are input.

12. **NOW TRY TO:**
 - a. Replace the five different variables by an array of five integer values.
 - b. Use a loop to input five array elements through the same function you created before.
 - c. Modify your averaging function to allow using this array as parameter (as a call by reference – see Lab 2 [example](#)).