Université d'Ottawa
Faculté de génie

École de science informatique
et de génie électrique

uOttawa
L'Université canadienne
Canada's university

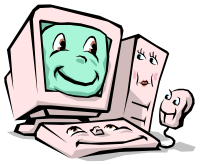
University of Ottawa
Faculty of Engineering

School of Electrical Engineering
and Computer Science

GNG 1106
Fundamentals of Engineering Computation

Lab 1 – Fall 2018

Prepared by Prof G. Abez
Revised by Prof. Y. Mao



Lab 1: Introduction to Programming in C

Before coming to the lab:

- Review the lab manual. It provides information on running CodeBlocks.
- Review the lab instruction document. It contains general information about labs throughout the term.
- Programming aspects to get familiarized with:
  - Editor/compiler and performing simple keyboard/screen input/output
  - Perform simple arithmetic calculations

A. Pair Programming

Pair programming is a technique used in industry where a program is developed by two persons. One person enters the code and acts as the driver and focuses on the tactical aspect of completing the task. The second person acts as an observer or navigator and considers the strategic direction of the programming. This 2nd person acts as a safety net for the driver. This allows the person observing to concentrate on logic which the person who is coding typically focuses more on the syntax of the program being entered. The roles of driver and observer are switched frequently. This approach has demonstrated that pair programming produces superior code with fewer defects (https://en.wikipedia.org/wiki/Pair_programming).

In the context of the lab, pair programming allows students to work together and learn from one another. Pairs of students should be made up of a student with no programming experience and one who has some programming experience. The student with experience will improve their knowledge and understanding by sharing their knowledge with someone with less experience. You should switch roles at each exercise.

B. Creating a C Program: Hello World (20 points)

For this part of the tutorial, the TA shall be giving instructions on creating and executing the following simple program. This process consists of three steps:

- Editing the program to create a C source file (with extension .c, as in hello.c)
- Compiling the C source file to create and executable file (with extension .exe, as in hello.exe)
- Execution of the executable file.

The person with less experience at programming should be at the keyboard (i.e. play the driver)

Hello World C Program:

```
#include <stdio.h>

void main()
{
    printf("Hello World\n");
}
```

See the lab manual to change the background/foreground colors to white background/black foreground.

C. Exercise: Temperature Conversion (80 points)

Switch roles (the driver becomes the observer and the observer becomes the driver).

Write a program that can convert a temperature in degrees Kelvin to degrees Fahrenheit. The user should be prompted to enter the temperature in degrees Kelvin and the program should print on the screen the temperature in degrees Fahrenheit.

In 1708, (Gabriel) Daniel Fahrenheit (Germany: 1686-1736) adopted a fixed point scale of temperature. The Fahrenheit scale sets the zero at the freezing point of a salt-water mixture and 100 at approximately body temperature. In 1742, Anders Celsius (Sweden: 1701-1744) described a mercury thermometer that had the zero of the scale set at the freezing point of pure water and 100 degrees set at the boiling point of pure water. The Celsius system is sometimes referred to as the centigrade system. The Kelvin scale, named after Lord Kelvin (William Thomson) (Scotland: 1824-1907), sets zero at the absolute zero of temperature (at which the motions of atoms and molecules practically stops) and uses degrees that are the same size as the Celsius system.

The Kelvin scale is related to the Celsius scale by the following relation: $TK = TC + 273.2$ where TK is the temperature in Kelvin and TC is the temperature in °C. The Celsius scale is related to the Fahrenheit scale by:
$TF = (TC * 9/5) + 32$ where TF is in °F.

The following table provides test cases that provides test cases that can be used for verifying your program.

Temperature Kelvin (TK) (input)	Temperature Celsius (TC) (TK - 273.2)	Temperature Fahrenheit (TF) (TC * 9/5 + 32)
0	-273.20	-459.76
250	-23.20	-9.76
300	26.80	80.24
400	126.80	260.24
1000	726.80	1,340.24

The `main` function:

Local Variables

- `degreeK`: for containing the Kelvin temperature.
- `degreeF`: to contain the Fahrenheit temperature.

Logic

- Prompt the user for a temperature in Kelvin with
  "*Please enter a temperature in degrees Kelvin:*"
- Read the value typed at the keyboard into the variable `degreeK`
- Call the function `convertKtoF`() to convert a value in Kelvin to Fahrenheit.
- Display the results with a message of the following form:
  "`Kelvin:  300 Fahrenheit: 80.24`"

The function `convertKtoF()`

Parameters

- `degreeK`: to receive the value of a temperature in Kelvin.

Local Variables

- `degreeC`: for computing the temperature in Celsius.
- `degreeF`: for computing the temperature in Fahrenheit.

Return Value: The value of the temperature in degrees Fahrenheit.

Logic

- Calculate the Celsius temperature and store its value in `degreeC` using: $TC = TK - 273.2$
- Calculate the Fahrenheit temperature and store its value in `degreeF` using: $TF = (TC * 9/5) + 32$
- Return the value in `degree`.