Lab 03: Temperature Conversion

Objective

This lab is to make you familiar with data types, assignment operator, arithmetic operations, and casting.

Useful links:
http://web.mst.edu/~cpp/cpp_coding_standard_v1_1.pdf (sec. 5, 6.1, and 6.2)
http://classes.mst.edu/compsci53/variabledeclarations.htm
http://classes.mst.edu/compsci53/assignmentandarithmeticoperators.htm

Assignment: Temperature Conversion

The temperature conversion program aims to convert the temperature value from Fahrenheit to Celsius. The formula for conversion is as follows:

$$({}^{\circ}F\ -\ 32)\ \mathrm{x}\ 5/9 = {}^{\circ}C$$

In the program the formula can be used as:

(tempInFahrenheit – 32) * CONVERSION_FACTOR = tempInCelsius

Where *CONVERSION_FACTOR = 5/9*

Write a program to calculate the temperature value in Celsius. The initial temperature value entered by the user *will be an integer*. Extend the program by type casting the calculated temperature to *int* using **static_cast<int>(temperature).** This casting will truncate the value after the decimal, not round it.

(Hint: Use **#include<math.h>** also in addition to **#include<iostream>)**

Program Structure:

```
//Include your headers. A common header usually needed is <iostream.h> to use cin and cout.
```

```
int main ()
{
  // variable declarations; say int variable1 = someNumericalValue1
  // read values from user
  // perform calculations
  // print output to user
  return 0;
}
```

Sample output

```
Welcome to the temperature converter!
Let’s convert from Fahrenheit to Celsius
Enter the temperature in Fahrenheit: 100
The temperature in Celsius is: 37
```

Things to consider to earn full grade

Your program will be graded based on the following:

- Use of constants where appropriate.
- Use of meaningful variable and constant names.
- Proper use of spacing for indentations.
- The program header (in which you specify the program description, your name, section, etc).
- Adequate commenting.
- Use of proper messages to prompt for input and labels to describe the output.
- Handling integer division where necessary.
- Use static casting where appropriate; **show that information lost was intentional**.
- **DO NOT** use any Library functions for rounding.
- Readability of the program
- Correctness of the program

Steps:

1. **Remotely connect to a Unix/Linux machine using Putty**

2. **Make a new directory named Lab3 under cs1580 folder and go into that directory**

3. **Identify the steps and design the program (try writing pseudocode for your reference)**

4. **Open a new C++ file and write the code*: jpico lab3.cpp***

5. **Compile the program: *fg++ lab3.cpp -o lab3***

6. **Run the program: lab3**

Some test cases:

	F → C
Fahrenheit	100
Celsius	37

7. **Submit your work**

Once you are sure you have the program running correctly, to submit a copy of your work, do the following command:

Section F (4:00 p.m. to 5:50 p.m.)**:** **cssubmit 1580 f 3**

Section G (6:00 p.m. to 7:50 p.m.)**:** **cssubmit 1580 g 3**

Note:

a) If you are unsure about submitting the code, show me/helpers the code and its output in order to avoid any mistakes or error.

b) When you submit your work, run your program for the set of input in the table above

8. **Logoff**