

**CSC112**                      **Spring 2011**  
**Fundamentals of Computer Science**  
**Lab 2**

In this lab, we will continue with Lab 1 (getting started with UNIX), and we will get started with learning the C++ language. Your assignment consists of two programs which progressively introduce new language features. Create a directory named Lab2. Keep all of your source and compiled programs in the directory Lab2.

### **Improving your Ubuntu/Linux working environment**

By default, a program called **bash** is the shell which interprets commands that you type at the command prompt. When you log-in to Ubuntu, bash reads a file named **.bashrc** for commands that define how your working environment will behave. One of the most irritating default settings is that the current working directory is not in the **PATH** list. Recall that **PATH** is a list of places (directories) where the system will look for an executable program that matches the command you have typed. If we write a program (or shell script) called “dostuff” when we type at the prompt:

```
student@ubuntu:~$ dostuff
```

we get the error message:

```
dostuff: command not found
```

The reason for this is that the current directory is not among the places that the system looks for the program named “dostuff”. Ok, we can type:

```
student@ubuntu:~$ ./dostuff
```

and the program runs as we expect. The reason for this is because we have specified a relative path to the program. Typing the “./” over and over for every program gets old quickly. We are going to fix our environment to include the current working directory in the **PATH**.

Edit your **.bashrc** file using vi. At the top of the file insert a line that reads:

```
PATH="$PATH:."
```

The **PATH** environment variable is a character string that contains a list of locations (directories); each location (directory) on the list is separated by a colon. The command we just added to the **bash** startup file puts the name of the present working directory, i.e., “.” into the **PATH**.

### **Getting Started with C++**

1. Our first program is pretty simple, just to get you writing on your own.

Write a C++ program to input a temperature in Fahrenheit and output the equivalent temperature in Celsius. The formula is:

$$C = 5/9(F - 32)$$

Be mindful of the distinction between integer division and floating point division.

2. Modify your program from part 1 to check for valid input. Use the member function `cin.fail()` to determine if the input is acceptable. If the input is unacceptable, print an error message to the error stream (using `cerr`), and exit.
3. Modify your program from part 2 to so that the temperature conversion calculation is done by a function called “fahrenheit\_to\_celsius()”. The function header should be:

```
double fahrenheit_to_celsius(double f)
```

**Turn in:** Change to the directory containing the sub-directory “Lab2” Create a file named “lab2.tar” using the command:

```
tar cf lab2.tar Lab2
```

Upload the file “lab2.tar” to your account on telesto.