

CS101 Lab-1

Today's Linux Commands

Recall how to open terminal and write commands from the previous tutorial. Recall that `.` represents the current directory and `..` represents the parent directory. `~` (tilde) is a shortcut for your home directory: `/home/<username>`.

- `cd ~` is a quick way to navigate back to your home directory
- `mkdir -p ~/lab1g1` tells mkdir to create all necessary directories if they don't already exist
- `cd ~/lab1g1` to change your working directory to lab1g1
- You can choose one of the following ways to write your programs. (Modify the filename according to your program name):
 - a. Type `gedit hello.c` to write your program. Save and **close** the editor. Go back to the blinking prompt on the terminal. (If you don't see the blinking prompt, you haven't closed gedit yet). Compile and run your program.
 - b. Type `gedit hello.c` to write your program. Save and **minimize** the editor. Go back to the terminal window and click on plus icon to start a new tab. Compile and run your program in the new tab.
 - c. Type `touch hello.c` to create the .c program file. Open gedit from the GUI (graphical user interface) with mouse (or your favorite text editor) and open the newly created file. Write and save your program. Go back to the terminal to compile and run your program.
- On the terminal, type `cat hello.c` to see the program contents
- To compile, type `gcc hello.c -o hello.out`
- To see the machine code, type `cat hello.out`
- To run the program, type `./hello.out` or its full path `~/lab1g1/hello.out`

Instructions:

1. Five programs named `keyboard.c`, `centigrade_to_fahrenheit.c`, (C) `prog3.c`, (D) `prog4.c`, and (E) `prog5.c` have been provided in this tutorial below. Type and run each of them to know how they work. Guess the intention behind (D) `prog4.c` and modify its print statement to provide expected output.

```
sizeof(long int);  
sizeof(long long int);  
sizeof(short);
```
2. After successfully working with the above five programs, write your own C program that does the following:
 - a. Read from the user the *base* and *height* of a right angled triangle.
 - b. Compute its area and print it to the user.
 - c. Choose appropriate data types for your program.
3. To run the programs, follow the steps discussed and mentioned in Tutorial-0.
4. Name your program and your variables appropriately. Write comments. Indent your code.

The Programs:

A) Reading data from the keyboard

```
// keyboard.c
#include <stdio.h> //keyboard.c
int main() {
    int num ;
    scanf("%d", &num) ;
    printf ("No. of students is %d\n", num) ;
}
```

B) Converting centigrade temperature to its Fahrenheit value

```
//centigrade_to_fahrenheit.c
//to compile this program, you can autocomplete its name with the help of TAB
key like this: centi<TAB>
#include <stdio.h>
int main() {
    float cent, fahr;
    scanf("%f", &cent);
    fahr = cent * (9.0 / 5.0) + 32;
    printf("%f C equals %f F\n", cent, fahr);
    return 0;
}
```

```
}
```

Can you guess what the following programs are doing?

C)

```
#include <stdio.h>
int main() {
    int n;
    scanf("%d", &n);
    printf("%d\n", n * n);
    return 0;
}
```

D)

```
#include <stdio.h>
int main () {
    int n;
    scanf("%d", &n);
    printf("%d\n", 1/n);
    return 0;
}
```

E)

```
#include <stdio.h>
int main() {
    int integerType;
    float floatType;
    double doubleType;
    char charType;
    // Sizeof operator is used to evaluate the size of a variable
    printf("Size of int: %ld bytes\n", sizeof(integerType));
    printf("Size of float: %ld bytes\n", sizeof(floatType));
    printf("Size of double: %ld bytes\n", sizeof(doubleType));
    printf("Size of char: %ld byte\n", sizeof(charType));
    return 0;
}
```