**CS 1050 Lab Assignment 1**

**SPRING 2017**

**\*\*\* FAILURE TO RETURN THIS DOCUMENT TO THE TA WILL RESULT IN A ZERO GRADE FOR THE LAB \*\*\***

**Directions :**

Complete the following lab assignment using the description given in each section.

**Purpose:**

🕐 Introduce basic Linux commands including the use of the vim editor.

🕐 Create a simple C program.

🕐 Login to tc.rnet server using putty.

🕐 Learn to compile, run and submit programs.

**Submission information:**

Submit this assignment by following the instructions given by your TA. SUBMIT ONLY the .c file (no a.out or executable file is required). All the lab assignments must be submitted before the end of the lab using the lab code given by the TA. Use the following submit command.  
  
$ submit <class> <assignment> <filename>  
eg: $ submit CS1050 LAB1 lab1.c

**Filename must be**: *labSection-lab1.c* (Include your respective lab section)

e.g. a-lab1.c

***Part 1: Putty***

**Description**:

Putty is a free piece of software used to connect to remote servers. Teaching Cluster is a server on campus which you will connect to using Putty or another telnet command console. (Macs will have their own “terminal” pre-installed.) If you are using a Windows computer at home, you can download Putty from Blackboard under Resources. There is no need to install Putty, just download it and double click to run it.

**Note:** All lab computers already have Putty, you do not need to download it. There are several options with this program, such as changing the font size and colors of the text, which you can feel free to explore on your own or ask your TA for help.

Once Putty is open, under the Host Name (or IP address) box type in:

**tc.rnet.missouri.edu**

Above is the address for the server, which we are connecting to. This will always be the same address. Once that is typed in, you can click Open or press enter. Don't change the default settings here or you will not be able to connect. Once you click Open, you will see *login as:* This is where you will type only your pawprint and press Enter. Then it should say yourpawprint@tc.rnet.missouri.edu *password:* Type in your password.

**IMPORTANT**: the cursor does not move when you type in your password. If you mess up, just hold backspace for a few seconds to clear it out and type it again. This is a security measure so that people can't figure out your password. Once logged in continue reading below. If you have problems, consult your TA. Whenever you have time please go through Resources tab under Blackboard.

***Part 2 : Linux Commands***

**Description:**

The following are a few of the important Linux commands needed to navigate the file system from a command console.

**pwd** – the command to show the current directory path

Usage: pwd

**ls** – the command to list out the contents of the current directory

Usage: ls

**mkdir** – the command to create a new folder at the current directory

Usage: mkdir *nameoffolder*

**rm** – the command to delete a file in the current directory

Usage: rm *nameoffile*

**cp** – the command to copy a file in the current directory

Usage: cp *nameoffiletocopy locationoffiletocopyincludingname*

**cd** – the command to change the current directory

Usage: cd *nameoffolder*

Note: The command “cd” will change the directory by moving up one level

**mv** – the command to move a file (can be used to rename a file)

Usage: mv *filename newfilename*

***Part 3: Using VIM***

**Description:**

Now in order to start creating our first program we will need a text editor. The text editor for this class is called VIM. While at first this text editor will seem challenging and different, once you get the hang of it you will be able to create programs and any text documents in fractions of the time than with another editor. TAs are going to show how to open, edit and save a file using VIM editor.

***Part 4: Compile and Run a C program***

Description:

Now open vim and create a file. Type the following program into the file, save it, and exit vim.

#include <stdio.h>

int main(void)

{

printf(“Hello World!\n”);

return 0;

}

Now to compile your first C program you will type in the following command.

gcc -Wall -Werror a-lab1.c

gcc is the name of the compiler that allows us to compile C programs and a-lab1.c is the name of the program we want to compile. Now that we have compiled our first program we run it by typing in *./a.out*. The ./ is the linux command to run an executable file, and a.out is the name of the executable file to run. Your program should now run and print out to the console “Hello World!”.

***Part 5: Debug a C program***

When you write a C program often you are not going to see the final result in first attempt. Your code might have errors. Errors can be broadly classified into two categories “Syntax errors” and “Logical errors”. Syntax errors generally happen when not following the correct C syntax (rules). GCC compiler is going to catch most of these syntax errors for you and also gives the line number where these syntax errors are happening. A program cannot be run before fixing the syntax errors so you need to fix all the syntax errors before you can see any type of output. Second type of error is “Logical errors” these are more difficult to fix as you can compile and run a code but the final output is incorrect. One way to find and fix the logical errors is to use GNU debugger. We are going to discuss about the debugger in the future labs or schedule a help session on “How to use a GNU debugger” but often using a simple printf statement can also help finding the logical errors.

Following C code computes the largest number of two numbers but it has syntax errors in it. TA is going to show what type syntax errors you will see and how to fix these errors.

include<stdio.h>

int main()

{

int i, Y;

printf("\n Enter the first number: ")

scanf("%d", x);

printf("\n Enter the second number: ");

scanf("%d", &y);

if(x>y)

{

Printf("\n is the largest number", x);

}

Else

{

printf("\n %d is the largest number", y);

}

printf("\n");

return 0;

}

***Part 6: Returning the lab document.***

After finishing the lab assignment, you are required to return the lab document to the TA. This is to ensure that students do their lab assignments in the lab and the lab document is not passed around to other labs/students. Returning the lab document and getting attendance is student’s responsibility so even if TA forgets to take the attendance or ask for the lab documents students should ask for their attendance and return the lab document to the TA.

**Sample Outputs**

Characters in bold are inputs from the user.

[sm3z5@babbage Lab1]$ ./a.out

Hello World!

Bonus part

Enter the distance: **0**

Distance should be a positive number

Enter the distance again: **1675**

Enter the speed: **-35**

Speed should be a positive number

Enter the speed again: **35**

Flight time of the plane is 107.678574 seconds

**Guidelines for Grading Lab 1**

**10 Points Possible (+0 points Bonus)**

**\*\*\* FAILURE TO RETURN THIS DOCUMENT TO THE TA WILL RESULT IN A ZERO GRADE FOR THE LAB \*\*\***

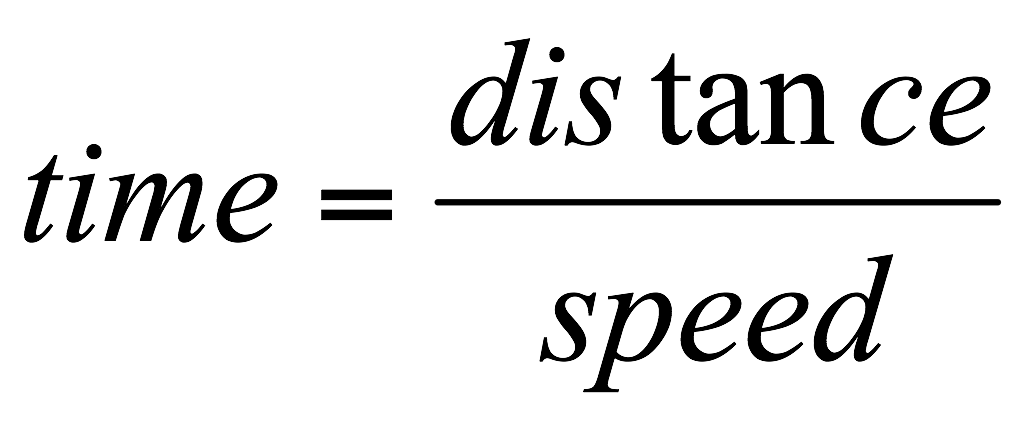
**General**

***If your program does not compile, or produce any input/output (I/O) because most of the source code is commented out then your lab will receive a grade of NO POINTS.*** In the first few labs the TAs will usually be able to inspect your code to see if what you submitted is close to the lab requirements and *may* award partial credit. However, for partial credit for future labs your C program must not only compile but also produce some valid I/O that meets the lab specifications.

**10 points:** Submitting the assignment on the blackboard.

**BONUS (0 points)**

For bonus part calculate the flight time of an airplane using the formula below.



Where the unit of time is seconds, speed is miles/ hr. and distance is meters.

Use the following in the calculation.

1 mile= 1600m (approx.)

1 hour=3600 seconds

Read the distance (in meters) and the speed (in miles/hr) of an airplane from the user and if they are negative display an error message and ask for the new input again. Assume that the user is going to give an invalid input only once so error check for input is only required once. After reading the inputs correctly from the user calculate the flight time of an airplane using the above formula and display the result.

Use floating point variables for distance, speed and time.

Check the sample outputs above to see how to gather information from the user, display error messages and the final output of the program.