Lab **3**

Variables

Learning Objectives

After completing this lab you will be able to do variable declarations and initializations with some more printing and math in C.

Basic Lab Instructions!

- Talk to your classmates for help.
- You may want to bring your textbook to future labs to look up syntax and examples.
- Stuck? Confused? Have a question? Ask a TA/Lab Engineer for help, or look at the book or past lecture slides.
- Complete as many problems as you can within the allotted time. You don't need to keep working on these exercises after you leave the lab.
- Before you leave today, make sure to check in with one of the Lab Engineers/TAs in the lab to get credit for your work.

Lab Tasks

Creating a New Project

After completing this section you will have created a project for today's lab.

1. Open the solution CS110Labs you created in the lab last week.
2. Add a new project called Lab02 within the solution CS110Labs. Set the project as startup project (right-click the project in Solution Explorer). STOP HERE! start with the lab tasks, and refer to the following steps accordingly.
3. Click File in the top menu bar of Visual Studio, and then select the Add New Item option (Add Existing Item option depending on the lab task).
4. Select C++ File
5. Type in the code listed.
6. Compile (Build Lab02) the program and execute it (Execute without Debugging).

Task #1: Variables and Names

You can print things out with printf and you can do math. The next step is to learn about variables. In programming a variable is nothing more than a name for something so you can use the name rather than repeating work done. They are also used to make the code more readable.

Open VariablesAndNames.c in your project Lab02. Read your .c file out loud, saying even the punctuation and symbols. Remember _ is an underscore character. We use this character a lot to put an imaginary space between words in variable names.

```c
#include <stdlib.h>
#include <stdio.h>

int main( void )
{
  int
  cars
  , drivers
  , passengers
  , cars_not_driven
  , cars_driven;

  float
  space_in_a_car
  , carpool_capacity
  , average_passengers_per_car;

  cars = 100;
  space_in_a_car = 4.0;
  drivers = 30;
  passengers = 90;
  cars_not_driven = cars - drivers;
  cars_driven = drivers;
  carpool_capacity = cars_driven * space_in_a_car;
  average_passengers_per_car = passengers / cars_driven;

  printf( "There are %d cars available.\n", cars );
  printf( "There are only %d drivers available.\n", drivers );
  printf( "There will be %d empty cars today.\n", cars_not_driven );
  printf( "We can transport %.1f people today.\n", carpool_capacity );
  printf( "We have %d to carpool today.\n", passengers );
  printf( "We need to put about %.1f in each car.\n", average_passengers_per_car );

  return EXIT_SUCCESS;
}
```

What You Should See

```
There are 100 cars available.
There are only 30 drivers available.
There will be 70 empty cars today.
We can transport 120.0 people today.
We have 90 to carpool today.
We need to put about 3.0 in each car.
```

What You Should Do on Your Own

Assignments turned in without these things will not receive any points.

1. Write a comment above each line explaining to yourself what it does in English.
2. Remember that 4.0 is a "floating point" number. Find out what that means.

3. I used 4.0 for space_in_a_car, but is that necessary? What happens if it's just 4?

You have to turn-in the modified VariablesAndNames.c file.

Task #2: More Variables and Printing

Every time you put " (double-quotes) around a piece of text you have been making a string. It is what you use to display on the console. Your program might print them, save them, send them, and many more.

In this exercise you'll learn how to make strings in C. As usual, just type this in, or use the provided MoreVariablesAndPrinting.c.

```c
#include <stdlib.h>
#include <stdio.h>

int main( void )
{
  char myName[]  = "Muaz Hashmi";
  char myEyes[]  = "Brown";
  char myTeeth[] = "White";
  char myHair[]  = "Black";

  int myAge    = 30; // not a lie
  int myHeight = 70; // inches
  int myWeight = 170; // lbs

  printf( "Let's talk about %s.\n", myName );
  printf( "He's %d inches tall.\n", myHeight );
  printf( "He's %d pounds heavy.\n", myWeight );
  printf( "Actually, that's not too heavy.\n" );
  printf( "He's got %s eyes and %s hair.\n", myEyes, myHair );
  printf( "His teeth are usually %s depending on the coffee.\n", myTeeth );

  // This line is tricky; try to get it exactly right.
  printf( "If I add %d, %d, and %d I get %d.\n"
  , myAge, myHeight, myWeight
  , (myAge + myHeight + myWeight) );

  return EXIT_SUCCESS;
}
```

What You Should See

```
Let's talk about Muaz Hashmi.
He's 70 inches tall.
He's 170 pounds heavy.
Actually, that's not too heavy.
He's got Brown eyes and Black hair.
His teeth are usually White depending on the coffee.
If I add 30, 70, and 170 I get 270.
```

What You Should Do on Your Own

Assignments turned in without these things will not receive any points.

1. Change all the variables so there isn't the my in front. Make sure you change the name everywhere, not just where you used = to set them.

2. Try to write some variables that convert the inches and pounds to centimeters and kilos. Don't just type in the measurements, but work out the math inside your C program.

```
Let's talk about Muaz Hashmi.
He's 70 inches (or 177.8 cm) tall.
He's 170 pounds (or 77.1107 kg) heavy.
Actually, that's not too heavy.
He's got Brown eyes and Black hair.
His teeth are usually White depending on the coffee.
If I add 30, 70, and 170 I get 270.
```

You have to turn-in the modified MoreVariablesAndPrinting.c file.

Task #3:Using Variables

Write a program UsingVariables.c that creates three variables: an int, a float, and a string.

Put the value 3 into the first variable, the value 2.71828 into the second, and the value "Computer Science" into the third. It does not matter what you call the variables.

Then, display the values of these three variables on the screen, one per line.

```
This is Lab 3
e is close to 2.71828
I am learning a bit about Computer Science
```

Your program should NOT look like this:

```
printf( "This is Lab 3\n" );
printf( "e is close to 2.71828\n" );
printf( "I am learning a bit about Computer Science\n" );
```

You must use three variables. You have to turn-in the UsingVariables.c file.

Task #4:Still Using Variables

Write a program StillUsingVariables.c that stores your name and year of graduation into variables, and displays their values on the screen.

Make sure that you use two variables, and that the variable type is the best for the variable. Also make sure that your variable names are good: they should always relate to its contents.

```
My name is Abdul and I'll graduate in 2016.
```

Notice that in the example above, the values "Abdul Hannan" and 2016 have been stored in variables before printing.

Your program should NOT look like this:

```
printf( "My name is Abdul Hannan and I'll graduate in 2016." );
```

You have to turn-in the StillUsingVariables.c file.

Task #5:Your Schedule

Write a program YourSchedule.c that use several variables to store the names of your classes and their teachers. Then, display a nice little table displaying your schedule.

It should roughly look similar to what I have done. I have used six variables for course names and six variables for teacher names. You should do the same.

```
+-------------------------------------------------------------+
| 1 |   Fundamentals of Computer Programming |       Mr. Anis |
| 2 |                    Fundamentals of ICT |     Mr. Jaudat |
| 3 |                   Discrete Mathematics |      Mr. Usman |
| 4 |                           Calculus-I |    Mr. Ramzan |
| 5 |                      Islamic Studies |       Ms. TBA |
| 6 | Communication and Interpersonal Skills |     Ms. Maria |
+------------------------------------------------------------+
```

You have to turn-in the YourSchedule.c file.

Hand in

Hand in the source code from this lab at the appropriate location on the blackboard system at LMS. You should hand in a single compressed/archived file named Lab_3_<your reg. No. XXX without angle brackets>.zip that contains ONLY the following files.

1. All completed C source files representing the work accomplished for this lab: VariablesAndNames.c; MoreVariablesAndPrinting.c; UsingVariables.c; StillUsingVariables.c; and YourSchedule.c. The files should contain author in the comments at the top.
2. An plain text file named OUTPUT.txt that includes a) author information at the beginning, b) a brief explanation of the lab, and c) any comments, or suggestions.

To Receive Credit

1. By showing up on time for lab, working on the lab solution, and staying to the end of the class period, only then you can receive full credit for the lab assignment.
2. Comment your program heavily. Intelligent comments and a clean, readable formatting of your code account for 20% of your grade.
3. In-class lab time is not intended as free time for working on your program assignments. Only if you have completely solved the lab assignment, including all challenges, and have had your work checked off for completeness by your TA/Lab Engineer should you begin the program assignment.