Lab Assignment - 1

Lab Assignment: Fundamentals of C

Objective:

1. How to create C programs?
2. How are C programs organized?
3. How to write a basic program to display text on the screen?
4. How to write program to perform basic arithmetic function?

What is C?

C is a ‘programming language’. That is, it is a language that allows you to specify exactly what you want your computer to do and explain clearly for other people what you are doing. C is a middle level multi-purpose programming language that can be run across several different platforms.

A middle level programming language uses keywords and symbols that humans are familiar with. High level programming languages were developed to make it easier for humans to program computers. Remember all the information inside a computer is in binary form, including the program instructions. It would be tedious for humans to enter this so-called low-level code, so different high level programming languages were developed, for example PASCAL and C, Java, Python etc.
A high-level language cannot run directly on your computer, it must be compiled (converted) into a set of simple instructions that a computer can use. Each instruction is used to instruct the computer to perform a specific operation.

C programming Environment:

C is a programming language of many different “dialects” because there are so many compilers. Each compiler is a little different. The library functions of one will have all of ANSI C, the standard, but will also contain additional functions.

Hello World Program:

C Program Code	**Description**

#include <stdio.h>
/**************************************/
/* Sample Program   */
/**************************************/
 	
int main ( )
{
    printf (“HELLO WORLD \n”);
    return 0;
}
	1. **Pre-Processor Directives** - used to tell the compiler of header files that have to be included, and to declare constants and global variables.
	2. **Programmer’s Block -** /* and */ are used to enclose pieces of text which describe what the program does. This text is called a comment.
	3. **main( )** is the start of the function main. The body of the function is enclosed between the curly brackets { }.
	4. **printf( )** is a function used to display a string of text on the screen

What did it all mean?

In C, programs are organized into blocks, called functions. A C function is like a mathematical function: it typically takes arguments; it performs an operation on these, and typically returns a result. For example, the addition of two numbers, a computer takes two numbers and performs the addition operation on them and then returns the result of the operation.

In C, the function called `main' is where the program execution starts. In the case of a very simple program, the lines which make a `main function' look like:

```
main()
{

}
```

However, this function doesn't do anything until instructions are put inside the curly brackets. In the example program above, there is one instruction inside:

```
printf(“HELLO WORLD \n”);
```

This is in fact an invocation of another function, printf. The printf function is provided with the compiler or operating system. In this example, it takes a single operand, which is a quoted string (“HELLO WORLD \n”) and displays this on the PC monitor.

Now look at the first line of code:

#include <stdio.h>

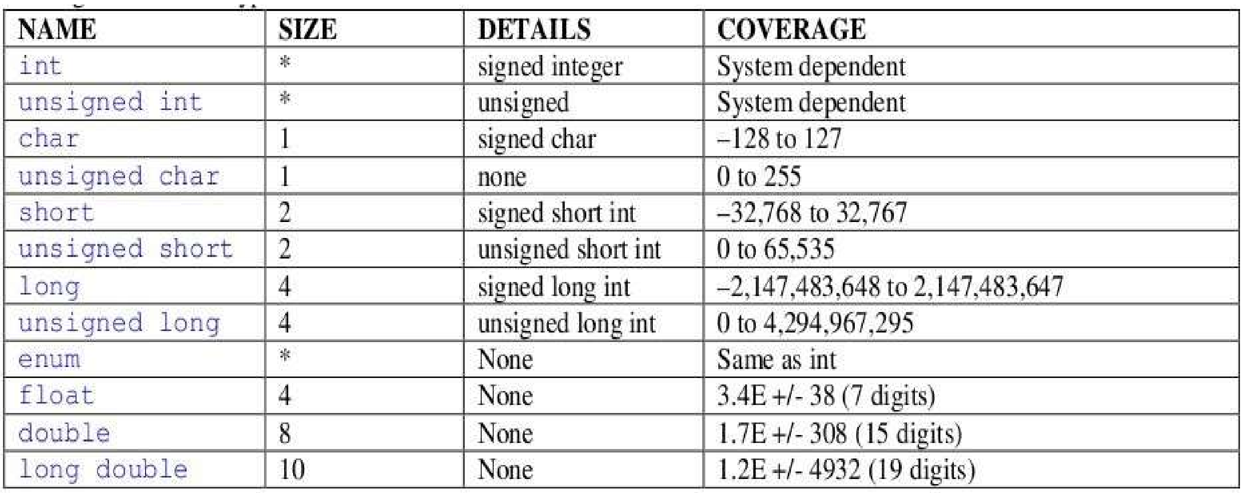
The # indicates that this is a pre-processor directive, which is an instruction to the compiler to do something before compiling the source code. If the main function is to call another function, in this case `printf', the pre- processor directive should instruct the compiler to ‘include’ in the program the file stdio.h. This file is called a header file, because it is usually included at the head of a program. It defines information about functions provided by a standard C library. The stdio.h header file contains declarations for the standard input and output functions available and must be included in the program to allow the use of the printf() function.

Declaring Variables:

A variable is a specific memory location set aside for a specific type of data and given a name for easy reference. The value stored in this memory location is variable, i.e. it can be changed. For example, if you are calculating the voltage across a fixed value of resistance as the current changes, the voltage variable would have a different value each time the current changed.

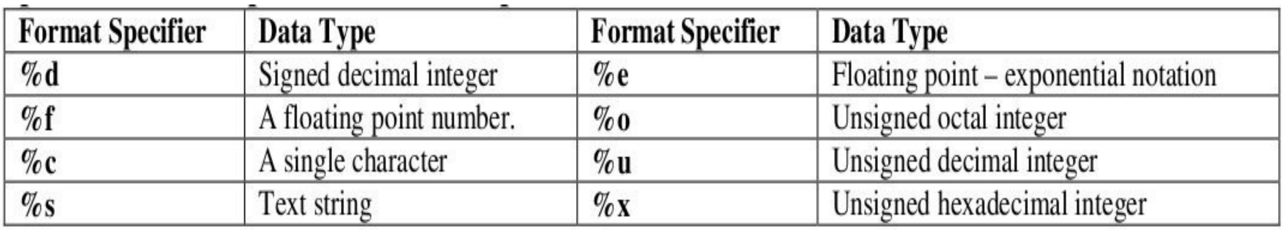
C statement to declare a memory location to store resistance values, where the value will be an integer (whole number), is as follows: **int resistance;**
C statement to declare a memory location to store current values, where the value will be a real number, is as follows: **float current;**
C statement to assign a value 200 to the variable called resistance is as follows:
resistance = 200;
C statement to assign a value 0.1 to the variable called current is as follows:
current = 0.1;
In C, all variables MUST be declared before they can be used in a program. To declare a variable, you must declare its type and identifier.

Data Types Range: C recognizes the data types shown in the table below.



NAME	SIZE	DETAILS	COVERAGE
int	*	signed integer	System dependent
unsigned int	*	unsigned	System dependent
char	1	signed char	–128 to 127
unsigned char	1	none	0 to 255
short	2	signed short int	–32,768 to 32,767
unsigned short	2	unsigned short int	0 to 65,535
long	4	signed long int	–2,147,483,648 to 2,147,483,647
unsigned long	4	unsigned long int	0 to 4,294,967,295
enum	*	None	Same as int
float	4	None	3.4E +/- 38 (7 digits)
double	8	None	1.7E +/- 308 (15 digits)
long double	10	None	1.2E +/- 4932 (19 digits)

printf and scanf Format Specifier:



Format Specifier	Data Type	Format Specifier	Data Type
%d	Signed decimal integer	%e	Floating point – exponential notation
%f	A floating point number.	%o	Unsigned octal integer
%c	A single character	%u	Unsigned decimal integer
%s	Text string	%x	Unsigned hexadecimal integer

The smallest location in memory which can be addressed is a byte, a byte consists of 8 bits, hence a byte can hold 256 possible numbers (2^8). An integer is normally 16 bits or 32 bits long. A float is normally 32 bits long. A char is 8 bits long Signed and unsigned are modifiers that can be used with any integral type. The char type is signed by default, but you can specify /J to make it unsigned by default.

For example, a variable of data type integer can be declared as follows:
int my_data;
This piece of code creates an integer variable called my_data.
To assign a value to this variable (load memory location) the C assignment operator (=) is used:
my_data = 10;
This line of code assigns the value 10 to the variable my_data. **Note an integer variable can only store integer values .**
A float is used to declare a floating point variable.

float more_data;
/* declares a floating point variable called more_data */ **more_data**
= 10.01;
/* more_data now assigned a value of 10.01. */

USER INPUT

The scanf( ) function.
The function scanf in stdio.h allows user data to be entered as the program is executing.
It has the following form **scanf(“ %f ”, &a);**
This will write floating-point data to the memory address where the variable is stored. Whenever & is placed in front of a variable it refers to the address where the data is stored.

In your laptop:

If you are using an Ubuntu OS

- Log in to the Ubuntu System with username and password.
- Open the terminal window.
- Create directory Lab-Sheet1 in your home directory (mkdir Lab-Sheet1).
- Make Lab-Sheet1 your working directory for Lab-Sheet1:
- Open gedit editor and type your C program and save it with .c extension.

In the Linux terminal **compile** the C program with gcc:

gcc -Wall lab1.c

The **`-Wall`** "switch" means "**W**arnings: **all**"; it instructs the **`gcc`** compiler to display all warning messages. If you find discrepancies, make changes, and recompile. If you have difficulty, consult a classmate or one of the Instructor. When you can compile with no errors, run the program with the **Linux command.**

./a.out

To create output file with different name you can compile it with:

gcc -Wall lab1.c -o lab1

then you can execute the object code

as: **./lab1**

Some Basic Linux Commands:

cd	ls	pwd	cal	rm	chmod
mkdir	cp	mv	date	cat	chown

If you are using a Windows OS

- Log in to the System with username and password.
- Open the Geany IDE.
- Create directory Lab-Sheet1 in your laptop.
- Create a new file using the template main.c.
- Type your C program and save it with .c extension.

Now click the compile, build, and execute button in the IDE to compile and execute the program.

Lab-Exercises:

1. Write a C Program to calculate the volume of a pool. The equation for determining the volume is

   Volume = length x width x depth. Assume that the pool has a length of 25 feet, a width of 10 feet and depth 6 feet.

2. Write a program to print Sum, Product and Quotient of two numbers 22 and 7.

3. Write a program to read the radius of sphere and compute its surface area and volume.

4. Write a program to input temperature in Celsius & to print its Fahrenheit equivalent. F = C * (9/5) + 32.

5. Write a program to read base & altitude of a triangle & print its area:

6. Write a program to calculate simple interest amount for deposit amount (P) kept in bank for (n) years at the rate of (r) simple interest per annum.

7. Write a program to input four numbers and print percentages of each numbers on their sum.

8. Write a program to evaluate the expression (i) a + b (ii) a – b (iii) a x b (iv) a/b (v) Remainder of a/b.

9. Write a program to input 5-digit integer number and print sum of digits in it.

10. Given a 5-digit integer number. Write a program to print it in reverse order (e.g. 92674 ⇒ 47629)

11. Write a program to compute:

$$F = \frac{2.9678 \text{ x } 10^{-27} + 0.876 \text{ x } 10^{-38}}{7.025 \text{ x } 10^{16} - 9.75 \text{ x } 10^{12}}$$

12. Write a *C* program to prompt the user for the *x*- and *y*-coordinates of the three corners of a triangle, calculate and print the lengths of the three sides, calculate and print the circumference of the triangle, and calculate and print the area of the triangle. The following mathematical formulas may be used in these calculations: –

    The formula for determining the length l_{AB} of the line between points (x_A, y_A) and (x_B, y_B) is

$$l_{AB} = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}\ .$$

    If l_{AB}, l_{BC}, and l_{CA} are the lengths of the three sides of a triangle, then the area of the triangle is

$$\sqrt{s \times (s - l_{AB}) \times (s - l_{BC}) \times (s - l_{CA})}$$

    where s is one-half the circumference of the triangle – i.e., s = ½ × ($l_{AB} + l_{BC} + l_{CA}$).