Course Code: CL118  
Credit Hour: 1  
Program Section: SE(B)19

PROGRAMMING FUNDAMENTALS LAB

Lab Instructors: Romasha Khurshid Nida Munawar

At the end of the lab, you will be able to understand:

* Variables
* Datatypes
* Format Specifiers
* Structure of Program
* compiler introduction
* Required libraries
* Printf, Scanf, commands
* Operators

**Variable:**

* C variable is a named location in a memory where a program can manipulate the data. This location is used to hold the value of the variable.
* The value of the C variable may get change in the program.
* C variable might be belonging to any of the data type like int, float, char etc.

**Declare and initialize variables:**

//DECLARATION

int balance; //creating a variable containing integer values

//DEFINITION

balance = 50; //assigning the value 50 in the balance variable

**Declare and initialize variables in 1-step:**

//DECLARATION and DEFINITION

int balance = 50;

**Name Your Variable: DO’s and Don’ts**

* lowercase and uppercase letters (characters) and digits
* do not use special characters’ like @ # & " ...
* do not use accented characters like é è à ...
* do not start with a digit
* start only with a letter
* spaces are forbidden
* \_ may be used instead of a space in the name of the variable

(YouCanUseUppercaseLettersBetweenWordsInsteadOfSpaces)

**Suggestion:**

It is advisable to be explicit about your variable name, choose the name that indicates what variable is all about and what it is used for so that people can understand easily.

**Data Types:**

* C data types are defined as the data storage format that a variable can store a data to perform a specific operation.
* Data types are used to define a variable before to use in a program.
* Size of variables are determined by data types.

# **1. Basic data types in C language:**

# **1.1. Integer data type:**

* Integer data type allows a variable to store numeric values.
* “int” keyword is used to refer integer data type.
* int (2 byte) can store values from -32,768 to +32,767
* int (4 byte) can store values from -2,147,483,648 to +2,147,483,647

**1.2. Character data type:**

* Character data type allows a variable to store only one character.
* Storage size of character data type is 1. We can store only one character using character data type.
* “char” keyword is used to refer character data type.
* For example, ‘A’ can be stored using char datatype. You can’t store more than one character using char data type.

**1.3. Floating point data type:**

Floating point data type consists of 2 types. They are,

* float
* double

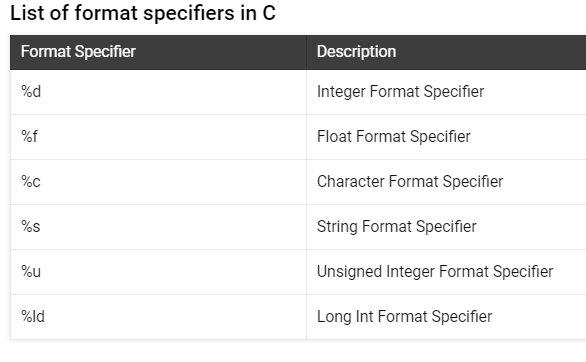
**1. float:**

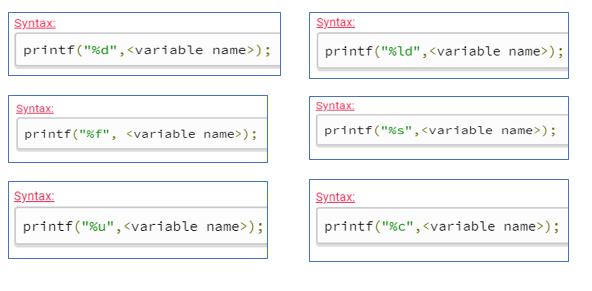
* Float data type allows a variable to store decimal values.
* We can use up-to 6 digits after decimal using float data type.
* For example, 10.456789 can be stored in a variable using float data type.

**2. double:**

* Double data type is also same as float data type which allows up-to 10 digits after decimal.

**Format Specifier:**





**Code Example for Format Specifier:**

**Use format specifier %d to print integer value:**

#include <stdio.h>

int main(void)

{

printf("If I have %d bills worth %d dollars each then I have %d dollars.",3,5,3\*5);

return 0;

}

**Use format specifier %C to print Character:**

#include <stdio.h>

int main(void) {

char letter, letter2;

printf("Please enter two letters: ");

scanf("%c%c", &letter, &letter2);

printf("I read the letters %c and %c.\n", letter, letter2);

return 0;

}

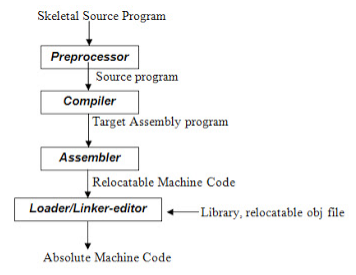
**Structure of Program (already discuss in Lab-02)**

**Description of Pre-Defined Syntax:**

|  |  |
| --- | --- |
| **C Basic commands** | **Explanation** |
| #include <stdio.h> | This is a preprocessor command that includes standard input output header file(stdio.h) from the C library before compiling a C program |
| int main() | This is the main function from where execution of any C program begins. |
| { | This indicates the beginning of the main function. |
| /\*\_some\_comments\_\*/ | whatever is given inside the command “/\* \*/” in any C program, won’t be considered for compilation and execution. |
| printf(“Hello\_World! “); | printf command prints the output onto the screen. |
| getch(); | This command waits for any character input from keyboard. |
| } | This indicates the end of the main function. |

#### **What is C compiler?**

* C Compiler is a program that converts human readable code into machine readable code. This process is called compilation.
* Human readable code is a program that consists of letters, digits and special characters that we type in program window. Machine readable code is in 0’s & 1’s
* For example, let’s assume that we type” HELLO” in program window. We know that we have typed “HELLO” in program window.
* But, processor knows only 01001000 for letter “H”, 01000101 for letter “E”, 01001100 for letter “L”, 01001100 for letter “L”, 01001111 for letter “O”
* Because, all C programs are executed by processor which is available in CPU.
* So, entire C source code should be converted into 0’s and 1’s as processor can understand only 0’s and 1’s.
* So, compiler converts entire source code into 0’s and 1’s during compilation.
* Output produced by compiler is in the form of 0’s and 1’s which is saved in .exe file. This file is called as executable or binary file.
* This binary file is executed by processor as per logic written in source code and the output is displayed in output window.



**printf and scanf**

* printf() and scanf() functions are inbuilt library functions in C programming language which are available in C library by default. These functions are declared and defined in “stdio.h” which is a header file in C language.
* We have to include “stdio.h” file as shown in below C program to make use of these printf() and scanf() library functions in C language.

**Note:**

C language is case sensitive. For example, printf() and scanf() are different from Printf() and Scanf(). All characters in printf() and scanf() functions must be in lower case.

**Operators in C**

* The symbols which are used to perform logical and mathematical operations in a C program are called C operators.
* These C operators join individual constants and variables to form expressions.
* Operators, functions, constants and variables are combined together to form expressions.
* Consider the expression A + B \* 5. where, +, \* are operators, A, B are variables, 5 is constant and A + B \* 5 is an expression.

# **Types of C operators:**

**C language offers many types of operators. They are,**

* Arithmetic operators
* Assignment operators
* Relational operators
* Logical operators (there are some more operator in C and will be discuss in coming labs)

**Arithmetic Operators:**

**BODMAS:**

BODMAS stands for Brackets, Ordinals (raised to the power of), Division, Multiplication, Addition and Subtraction.

**BODMAS Format:**

* Anything in brackets gets calculated first
* Order (or Exponent) gets calculated next
* Please remember that Division and Multiplication are in the same group meaning that if you have a formula with both division and multiplication the calculation is performed from left to right. But if you have a Division and Addition is in the formula then Division will be done before Addition.

**Code Example for Arithmetic operator:**

#include <stdio.h>

int main() {

printf("3+2 equals %d and 3-2 equals %d and 3\*2 equals %d\n", 3+2, 3-2, 3\*2);

printf("3+2\*3 equals %d and (3+2)\*3 equals %d\n", 3+2\*3, (3+2)\*3);

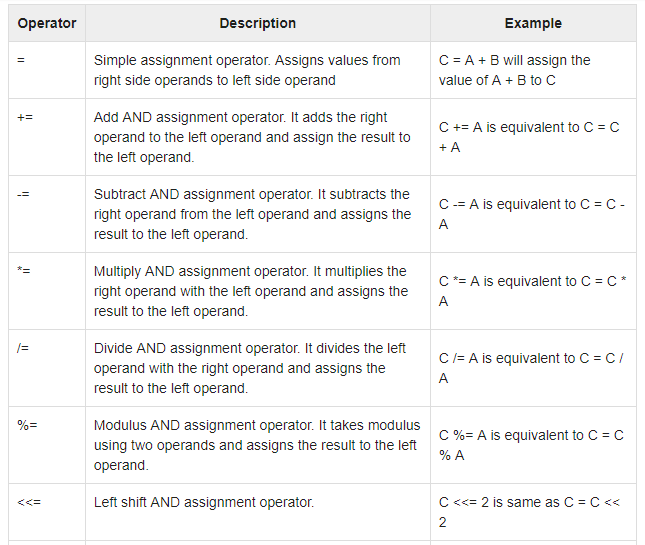
printf("2\*8-2\*7-4 equals %d\n", 2\*8-2\*7-4);

return 0;

**}**

**Assignment operators**

An assignment operator is used for assigning a value to a variable. The most common assignment operator is =



**Relational Operator:**

A relational operator checks the relationship between two operands. If the relation is true, it returns 1; if the relation is false, it returns value 0.

Relational operators are used in [decision making](https://www.programiz.com/c-programming/c-if-else-statement) and [loops](https://www.programiz.com/c-programming/c-for-loop).

| Operator | Meaning of Operator | Example |
| --- | --- | --- |
| == | Equal to | 5 == 3 is evaluated to 0 |
| > | Greater than | 5 > 3 is evaluated to 1 |
| < | Less than | 5 < 3 is evaluated to 0 |
| != | Not equal to | 5 != 3 is evaluated to 1 |
| >= | Greater than or equal to | 5 >= 3 is evaluated to 1 |
| <= | Less than or equal to | 5 <= 3 is evaluated to 0 |

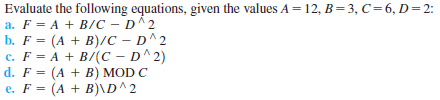
**Logical Operator**

An expression containing logical operator returns either 0 or 1 depending upon whether expression results true or false. Logical operators are commonly used in [decision making in C programming](https://www.programiz.com/c-programming/c-if-else-statement).

| Operator | Meaning | Example |
| --- | --- | --- |
| && | Logical AND. True only if all operands are true | If c = 5 and d = 2 then, expression ((c==5) && (d>5)) equals to 0. |
| || | Logical OR. True only if either one operand is true | If c = 5 and d = 2 then, expression ((c==5) || (d>5)) equals to 1. |
| ! | Logical NOT. True only if the operand is 0 | If c = 5 then, expression !(c==5) equals to 0. |

**Lab Activities:**

1. Write aC Program to convert temperature from Fahrenheit to Celsius
2. Let’s suppose your university organizes a free medical assessment after every two months in which your height and weight have been recorded, now Write a C program that takes your height, weight and age as run-time input ?
3. Print the Output:



1. Run Following expressions in your compiler

* printf("2\*(8-2\*(7-4)) equals %d\n", 2\*(8-2\*(7-4)));
* printf("2\*(8-2\*7)-4 equals %d\n", 2\*(8-2\*7)-4);

1. Write a program in C which takes distance in kilometers and convert it into meter. (Use proper names of variables)
2. Write a program that asks user to give number of days as input and convert it into weeks and days.
3. Write a program with finds the area and perimeter of a rectangle

**Relevant Formula:**

**Area of Rectangle = L X W**

**Perimeter of Rectangle = 2 X (L + W)**

1. You work for the Independence day. You've been asked how much time is left until the Procrastination day (march 25th).

Given that you've been asked on August 23th, please write a C-program which performs arithmetic in order to produce the following output:

Dear Procrastinator,  
You still have to wait for X days (Y minutes or Z seconds) before you can procrastinate!

Here, X is the remaining number of days (25-23), Y is the number of minutes (60 \* 24 \* X) and Z is the number of seconds (60 \* Y). The sentence has to be exactly the one displayed above, replacing X, Y and Z with the computed values. The format has to be followed precisely.

*Warning:*You cannot simply perform these calculations yourself and print the values - your program must calculate them and print them using the %d format specifier.

1. Write aC program to swap two numbers using a temporary variable?
2. **-**Write aC program to swap two numbers without using a temporary variable?