JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY, GUNA

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Course: Computer Programming Lab

Course Code: CS201

B. Tech. I Sem. (CSE, ECE, MECH, CE, CHE)

Lab-5

Aim: Getting started with C-language using Input-output functions

C Language: C is a powerful general-purpose high level computer programming language. It can be used to develop software like operating systems, device drivers, compilers, embedded systems, gaming, and scientific computing etc. Programming in C language is an excellent platform for the beginners. Basic features of this language are:

- **Case Sensitivity:** C is **case sensitive** language. All commands must be in **lowercase** only.
- **Header files:** Program starts with **header files** (also known as preprocessor directives). All programs requires **#include<stdio.h>** header file to import predefined input/output (printf, scanf etc.) functions in the program from the C language library. Other header files (#include<conio.h, #include<math.h> etc.) can also be included as per requirements.
- **main function:** Starting point of all C programs is identified by the word **main ( ). void main( )** and **int main ( )** are also used in place of main ( ). Modern compiler mostly uses **int main ( ).** Parentheses followed by keyword **main** indicate that no arguments supplied to the program.
- **Curly braces:** The two curly braces, **{ and },** signify the beginning and end of a program. All the program statements must be written within these two brackets.
- **Semicolon:** Each statement must be terminated with a **semicolon (;)**. Multiple statements separated with semicolon can be in the same line. White space is ignored. Statements can continue over many lines.
- **Keywords:** These are the reserved/predefined words in any programming language. There are 32 keywords in C language such as int, float, char, if, for, while etc. All keywords are written only in lowercase letters.
- **Data types:** In C programming, data types are declared with variables. This determines type and size of data associated with variables. Four basic data types are **integer** (keyword: **int**), **floating/fractional (**keyword: **float), character (**keyword: **char),** and **string (**keyword: **char [length of string]).** Group of multiple characters is known as string.
- **Variables:** These are the **user defined** names to hold the value of a data type. These can be made up of letters (lower and upper case) and digits. **Names must not begin with digits and special characters.** Only one special character **underscore ("_")** can be use as variable name. Same name defined in upper case and lower case is different. For example: Variable name '**A**' is not same as '**a**'.

Execute following program examples one by one to learn basic programming concepts.

Example#1:

```
/*Comment: Basic Structure of C language program*/
#include <stdio.h>        //Preprocessor or header file
int main()                //Declaration of function
{                         //Opening bracket
                          //Program statements are written in this area
return 0;                 //informs operating system that the program has completed without any errors.
}                         //Closing bracket
```

Example#2:

```
/* Syntax to use prinf function, space key, \t, \n and \" */
#include <stdio.h>
int main()
{
   printf("Hello, Jaypee University");                    //Syntax for printf function
   printf("\tHello \t Jaypee \t University");            // Use of horizontal tab character \t
   printf("\nHello\n Jaypee \n  University");             // Use of new line character \n
   printf("\n\"Hello\"\n\"Jaypee\"\n\"University\"");     //Use of new double quote character "\
return 0;
}
```

Example#3:

```
/* Syntax to take integer user inputs using scanf function */
#include<stdio.h>
Int main ( )
{
  int A;                                  //Declaration of integer data with variable name A
  printf("Enter vaue of A: ");            //User comment to enter value of variable A
  scanf("%d", &A);                        //scanf function to take user input of integer data
  printf("The value of A is %d", A);      //printf function to print value of A
return 0;
}
```

Example#4:

```
/* Syntax to take float user inputs using scanf function */
#include<stdio.h>
Int main ( )
{
  float ht;                               //float data declaration with variable name ht
  printf("Enter value of ht: ");          //User comment to enter value of ht
  scanf("%f", &ht);                       //scanf function to take user input of float data
  printf("Entered character is=%f", ht);  //printf function to print value of ht
return 0;
}
```

Example#5:

```
/* Syntax to take character user inputs using scanf function */
#include<stdio.h>
Int main ( )
{
   printf("Enter character Ch: ");              //User comment to enter value of Ch
   scanf("%c", &Ch);                            //scanf function to take user input of character data
   printf("Entered character is=%c", Ch); /     /printf function to print value of Ch
return 0;
}
```

Example#6:

```
/* Syntax to take string user inputs using scanf function */

#include<stdio.h>
Int main ( )
{
   char str[10];                          //string data declaration with variable name str with length 10
   printf("Enter string : ");             //User comment to enter value of str
   scanf("%s", &str);                     //scanf function to take user input of character data
   printf("The value of a is=%s", str);   //printf function to print value of str
   return 0;
}
```

Example#7:

```
/* Syntax to use gets and puts string functions */

#include<stdio.h>
Int main ( )
{
   char str[10];                          //string data declaration with variable name str with length 10
   puts("Enter string : ");               //User comment to enter value of str
   gets(str);                             //scanf function to take user input of character data
   puts(str);                             //printf function to print value of str
   return 0;
}
```

Note: Input function **gets** and output function **puts** can handle only string data type. Input function **scanf** scans string upto only first white space while **gets** scans entire string including white spaces. For Example: If string user input 'Computer Programming' is taken by **scanf** only 'Computer' will be printed while **gets** function will print complete string 'Computer Programming'.

Exercise#1: Write C language program to perform following operations using output function **printf**:

(a) Print A to Y in a single line.

(b) Print following pattern using **space key** and new line characters **\n**.

```
    A
   BCD
  EFGHI
 JKLMNOP
QRSTUVWXY
```

(c) Rotate above pattern vertically (first line in last line and vice versa) and apply double quote in each line using **double quote character \".**

Exercise#2: Write C language program to print following table using output function **printf**, horizontal tab character **\t** and new line character **\n**.

S. No.	Name	Age in Years	Height in Feet	Gender
1	Ram	21	6.2	M
2	Seeta	20	5.6	F

Exercise#3: Write **user input** C language program to print your **Name**, **Age in Years**, **Height in feet** and **Gender** using format specifier **%s** for string (name) inputs, **%d** for integer (age) inputs, **%f** for floating (height) inputs, **%c** for character (gender) inputs respectively. Use input function **scanf** to take user inputs. Show all four inputs in different line by using new line characters **\n**.

Exercise#4: Write **user input** C language program to repeat **Exercise#2.**

Exercise#5: Write **user input** C language program to print your Name, Branch name, University name, address of University by using string input function **gets,** string output function **puts** and new line characters **\n**.