

Birla Institute of Technology & Science, Pilani, Hyderabad Campus
First Semester 2020-2021

Computer Programming [CS F111] Lab 2

I. Introduction to scanf() to read inputs from the users

- **scanf()** function is used to accept the input from the user through keyboard.
- Syntax
 - `scanf("control string", &var1, &var2... &varN);`
- Variable name is prefixed with **&** (ampersand) symbol. It is known as "Address Of" operator and specifies a memory location to the scanf function where the value associated with variable will be stored in the computer's memory. The "&" symbol in scanf() is mandatory.

Note: Missing the ampersand in scanf() is the most common C Programming error because in this case, the compiler will not produce any compile time error.

- **Control String:** It contains field specifications which direct the interpretation of input data. It consists of conversion character % and type specifier. It is written in double quotes. Various type specifiers are listed below. Commonly used control strings are:

%d	integer
%f	float
%c	char

II. Introduction to printf() to read inputs from the users

- **printf()** function is used to print/output the variables' values to the user's terminal/display.
- Syntax
 - `printf("control string", var1, var2... varN);`
- Observe that there is no **&** (ampersand) symbol associated with variables. If we use the "Address Of" operator, printf function prints the memory (stored address) location of the associated variable. In simple terms, use the "&" symbol (prefixed to the variable name) in case if you want to print the address of the variable, and do not use it if you want to print the value associated with the variable.

III. Introduction to conditional constructs

- The **if** statement enables us to test for a condition (such as whether two variables are equal) and branch to different parts of the code, depending on the result of the conditions constructed using relational and logical. The simplest form is the following:

```
if (TestExpr) //TestExpr: Test Expression
{
    block of statements executed when "if"
    condition is true;
}
```

- Test expression can either have a relational operator(&&, ||, !) or a logical operator (==, >, <, >=, <= !=) but eventually evaluate to a boolean value, true (represented as **non-zero** value) or false (represented as **0**).
- The **if..else** statement enables us to test for a condition (such as whether two variables are equal) and branch to different parts of the code, depending on the result of the conditions constructed using relational and logical. The simplest form is the following:

```
if (TestExpr) //TestExpr: Test Expression
{
    block of statements executed when "if"
    condition is true;
}
else
{
    block of statements executed when "if"
    condition is false;
}
```

- The **if..else if..else** statement enables us to test multiple conditions (such as to check whether a number is in any of the multiple ranges) and branch to different parts of the code, depending on the result of the conditions constructed using relational and logical. The simplest form is the following:

```
if (TestExpr1) //TestExpr1: Test Expression1
{
```

```

        block of statements executed when "if"
        condition is true;
    }
    else if(TestExpr2) //TestExpr2: Test Expression2
    {
        block of statements executed when above "if"
        condition is true and previous if condition
        (corresponds to TestExpr1) is false;
    }
    else
    {
        block of statements executed when above "if"
        condition (corresponds to TestExpr2) is false;
    }

```

- You can also nest above conditional statements
- Conditional statements in detail will be discussed in upcoming labs

IV. Problems:

1. Try to achieve the following design using printf(). Notice the space between the characters.

```

#
#   #
#   #   #
#   #   #   #

```

Code:

```

#include<stdio.h>
int main()
{
    printf("#\n");
    printf("#\t#\n");
    printf("#\t#\t#\n");
    printf("#\t#\t#\t#\n");
    return 0;
}

```

2. Given values a, b, c and x, write a program to compute and display the value of the following equation.

$$y = ax^3 + bx^2 + cx + 5$$

Code:

```
#include<stdio.h>
#include<math.h>
int main()
{
    int a = 2 ,b = 3,c = 4,x = 2,y;
    y = a*pow(x,3) + b*pow(x,2) + c*x + 5;
    printf("%d", y);
    return 0;
}
```

3. Write a program to take 2 integers as input and print the largest of the two. Consider that the values of 2 integers are different. Your program should only use **if** conditional statements.

```
#include<stdio.h>
int main()
{
    int num1, num2;
    printf("Enter two integers:\n");
    scanf("%d",&num1);
    scanf("%d",&num2);
    if(num1 > num2)
    {
        printf("%d\n",num1);
    }

    if(num2 > num1)
    {
        printf("%d\n",num2);
    }

    return 0;
}
```

4. Write a program to take 2 floating point numbers as input and print if the two numbers are equal or less than one another or greater than one another. Your program should use **nested if..else** conditional statement(s).

```

#include<stdio.h>
int main()
{
    float num1, num2;
    printf("Enter two floating point numbers:\n");
    scanf("%f %f",&num1,&num2);

    if (num1 >= num2)
    {
        if (num1 == num2)
        {
            printf("Result: %f = %f",num1,num2);
        }
        else
        {
            printf("Result: %f > %f", num1, num2);
        }
    }
    else
    {
        printf("Result: %f < %f",num1, num2);
    }

    return 0;
}

```

5. Test the limits of signed and unsigned integers

```

#include<stdio.h>
#include<limits.h>
int main()
{
    unsigned short x = USHRT_MAX;
    unsigned short y = SHRT_MAX;
    printf("Max unsigned short value: %u\n",x);
    x = x+1;
    printf("Max unsigned short value + 1: %u\n",x);
    printf("Max short value: %d\n",y);
    y = y+1;
    printf("Max short value + 1: %d\n",y);
    return 0;
}

```

V. Exercises:

1. In the above Section IV, third program is utilizing two **if** conditional statements. Modify the program to use only single **if** conditional statement and no other conditional statements.
2. From the above Section IV, write the 3rd program using single **if..else** conditional statement.
3. From the above Section IV, write the 4th program using **if..else if..else** conditional statement(s).
4. Relationship between speed, time, acceleration and distance is governed by the following formula. Given speed (**u**), time (**t**) and acceleration (**a**), find the distance (**s**). Consider **u** and **t** to be integers but **a** and **s** to be floating point numbers.

$$s = ut + \frac{1}{2} at^2$$

NOTE: Upload the screenshots of the Exercise programs along with the displayed results into your corresponding Google Classroom.

PATH to Submit the Screenshots:

Google Classroom --> Classwork --> View Assignment --> Create/Upload files

***** GOOD LUCK *****