

# **F. Y. B. B. A. (C.A.) Semester I**

## **Lab Book**

Name: \_\_\_\_\_

College Name: \_\_\_\_\_

Roll No. : \_\_\_\_\_

Division: \_\_\_\_\_

Academic Year: \_\_\_\_\_

**CA-106:**  
**Computer Laboratory Based on**  
**CA-103 and Add-on (PPA)**

**Section I**

**‘C’ Programming**

## Assignment Completion Sheet

### Index Section-I (C Programming)

Sr.No	Title	Signature
1	Data types and operators	
2	Managing Input and Output	
3	Decision Making using if and if-else	
4	Decision Making using Switch	
5	Loop Control structures	
6	Nested Loops	
7	Demonstration of 1-D Arrays	
8	Demonstration of 2-D Arrays	
9	Strings	
10	Functions	
11	Pointers and Structure	

Name and Signature of Subject Teacher

Head of Department

Date:-

## Assignment Completion Sheet

### Index Section-II (PPA)

Sr.No	Title	Signature
1	DOS Commands and MS Office	
2	Scratch Programming	
3	Mini Project using Scratch	

Name and Signature of Subject Teacher

Head of Department

Date:-

# **Introduction**

## **About the Book**

This workbook is intended to be used by FYBBA (CA) students for their practical purpose. It helps to the students for clearing their theoretical as well as practical concepts.

### **Instructions to the Students and Instructors:**

- Students should carry workbook while coming to the practical.
- Students should complete all the practical assignments within given time interval.
- Instructors should check all the assignments regularly and guide to the students accordingly.
- Students have to solve exercises of Section I (C Programming) and Section II ( PPA) but there will not be any evaluation for Section II from University side.
- There is mini project on Scratch that student have to develop and get it checked from subject teacher

### **Editors:**

**1.Mr. Satyavan M. Kunjir**

**2.Mr. Yogesh Ingale**

**3.Mrs. Deepashri Mehendale**

**4.Mrs. Reshma Masurekar**

## Exercise 1: Data types and operators

### ❖ Data Types in C:

Each variable in C has an associated data type. Each data type requires different amount of memory and has some specific operations which can be performed over it. Let us briefly describe them one by one:

Following are the examples of some very common data types used in C:

1. **char:** The most basic data type in C. It stores a single character and requires a single byte of memory in almost all compilers.
2. **int:** As the name suggests, an int variable is used to store an integer.
3. **float:** It is used to store decimal numbers (numbers with floating point value) with single precision.
4. **double:** It is used to store decimal numbers (numbers with floating point value) with double precision.

Different data types also have different ranges up to which they can store numbers. These ranges may vary from compiler to compiler. Below is list of ranges along with the memory requirement and format specifiers on 32 bit gcc compiler.

DATA TYPE	MEMORY (BYTES)	RANGE	FORMAT SPECIFIER
short int	2	-32,768 to 32,767	%hd
unsigned short int	2	0 to 65,535	%hu
unsigned int	4	0 to 4,294,967,295	%u
int	4	-2,147,483,648 to 2,147,483,647	%d
long int	4	-2,147,483,648 to 2,147,483,647	%ld
unsigned long int	4	0 to 4,294,967,295	%lu
long long int	8	$-(2^{63})$ to $(2^{63})-1$	%lld
unsigned long long int	8	0 to 18,446,744,073,709,551,615	%llu

DATA TYPE	MEMORY (BYTES)	RANGE	FORMAT SPECIFIER
signed char	1	-128 to 127	%c
unsigned char	1	0 to 255	%c
Float	4		%f
Double	8		%lf
long double	12		%Lf

We can use the sizeof() operator to check the size of a variable.

### ❖ Operators in C:

An operator is a symbol that tells the compiler to perform specific mathematical or logical functions. C language is rich in built-in operators and provides the following types of operators.

1. Arithmetic Operators (+, -, \*, /, %)
2. Relational Operators (<, <=, >, >=, ==, !=)
3. Logical Operators (&&, ||, !)
4. Bitwise Operators(&, |, ^, <<, >>, ~)
5. Assignment Operators (=)
6. Increment Operator(++ ) and Decrement Operator(--)
7. Conditional Operator (expr1?expr2:expr3)

### ❖ Assignments based on Data Types and Operators:

**(Hint: Do not accept input from user. Declare variables with default values)**

- 1) Write a C Program to demonstrate the working of arithmetic operators (Associativity and precedence of arithmetic operators is expected)
- 2) Write a C program to find maximum of two numbers using conditional operator.
- 3) Write a C Program to find maximum of three numbers using logical operators.

- 4) Write a C Program to display Quotient and Remainder of division of two variable.
- 5) Write a C Program which illustrate increment and decrement operators (Use of Pre and Post increment is expected)
- 6) Write a program to display the size of different data types.
- 7) Write a program to swap the values of two variables using bitwise operator (^)
- 8) Write a C program which illustrate the use of Bitwise And, Bitwise Or and Bitwise XOR Operator).
- 9) Write a C program which illustrate Bitwise Left shift and Right Shift operators.
- 10) Write a C program to check Least Significant Bit (LSB) of a number is set or not.
- 11) Write a C program to check Most Significant Bit (MSB) of a number is set or not.
- 12) Write a C program to flip bits of a binary number using bitwise operator.
- 13) Write a C program to check whether a number is even or odd using bitwise operator.

Signature of the instructor

Date



## Exercise 2: Managing I/O Operations

Many functions for input and output purpose have been provided in the C standard input and output library. These functions interact with the standard input usually the keyboard and the standard output usually the screen.

### ❖ Input function scanf():

The general purpose input function is scanf() used to supply the values of variables through keyboards.

Syntax: scanf("control string",&var1,&var2....,&varn);

Where var1, var2 and varn are variable name.

The control string may contain format specifiers such as (%d %f %c %s etc).

%d- for integer values

%f- for float values

%c- for character values

Eg. scanf("%d",&a) ;

scanf("%d %f %c",&a,&b,&c);

### ❖ Output function printf():

printf() is a output function which is used to print on screen the value contained in a variable.

Syntax: printf("Control String", var1, var2....varn);

The control string may contain escape sequence characters (\n,\t,\v,\b etc) and format specifiers (%d %f %c etc).

Ex. printf("C is very easy");

printf(" The value of a=%d",a);

printf("\n The value of b=%d and sum=%f",b,sum);

### ❖ Character input and output function:

C provides functions for reading and writing one character at a time by using getchar() and putchar() function respectively. The function getchar() reads and return an input character from the standard input device.

Syntax: variable=getchar();

Eg. char ch;

ch=getchar();

The function putchar() writes a single character on the standard output device.

Syntax: putchar(variable);

Eg. putchar(ch);

### ❖ Assignments based on Input and Output functions:

1. Accept dimensions of a cylinder and print the surface area and volume.
2. Accept temperatures in Fahrenheit (F) and print it in Celsius(C) and Kelvin (K) (Hint:  $C = 5.0/9(F-32)$ ,  $K = C + 273.15$ )
3. Accept initial velocity (u), acceleration (a) and time (t). Print the final velocity (v) and the distance travelled (s) (Hint:  $v = u + at$ ,  $s = u + at^2$ )
4. Accept two numbers and print arithmetic and harmonic mean of the two numbers (Hint:  $AM = (a+b)/2$ )

,HM = ab/(a+b) )

5. Accept three dimensions length (l), breadth(b) and height(h) of a cuboid and print surface area and volume (Hint : surface area=2(lb+lh+bh ), volume = lbh )
6. Accept a character from the keyboard and display its previous and next character in order. Ex. If character entered is 'd', display "The previous character is c", "The next character is e".
7. Accept a character from the user and display its ASCII value.
8. Accept the x and y coordinates of two points and compute the distance between the two points.
9. Accept two integers from the user and interchange them. Display the interchanged numbers.
10. A cashier has currency notes of denomination 1, 5 and 10. Accept the amount to be withdrawn from the user and print the total number of currency notes of each denomination the cashier will have to give.

Signature of the instructor

Date

### Exercise 3: Decision Making using if and if-else

Sometimes it is necessary to alter the sequence of execution of statements based on certain conditions or we may require some statements to be executed repeatedly until some condition is met, this involves decision control structure and loop control structure respectively.

#### ❖ Decision Control Structure:

Many programs require testing of some conditions at some point in the program and selecting one of the alternative paths depending upon the result of the condition. This is known as decision making.

C provides 3 decision making statements

1. if statement
2. if else statement
3. switch

#### 1. if statement:

This is the simplest form of decision making statement in C. it allows decision to be made by evaluating an expression. Depending upon the result (true or false) the program execution proceeds.

Syntax:

```
if(expression)      or      if(expression)
    statement;        {
                        Statements;
                        }
```

The keyword if must be followed by a set of parentheses containing a single expression to be tested. The statement is executed only if the expression is true. If the condition evaluates to false, the statement is skipped.

Ex. following program accepts two numbers and display maximum number.

```
#include<stdio.h>
main()
{
    int a,b;
    printf("\n Enter any two numbers=");
    scanf("%d%d",&a,&b);
    If(a>b)
        printf("\n %d is maximum",a);
    If(b>a)
        printf("\n %d is maximum",b);
    If(a==b)
        printf("\n both numbers are same");
}
```

#### 2. if-else statement:

The if statement will execute the statement if the expression is true otherwise it will be skipped. However in many cases we require an alternate statement to be executed if the expression evaluates to false. This is possible using if else statement.

Syntax:

If (expression)	or	if(expression)
Statement 1		{
else		statement part 1
Statement 2		}
		else
		{
		Statement part 2
		}

Here, expression 1 is executed if expression is true otherwise statment2 is executed.  
Thus either statment1 or statment2 will be executed, never both.

Ex. following program accepts a number and display whether it is even or odd

```
#include<stdio.h>
main()
{
int n;
printf("\n Enter any number=");
scanf("%d",&n);
if( n%2==0)
printf("\n %d is even number",n);
else
printf("\n %d is Odd number",n);
}
```

Following is a program to accept a year as input to check whether it is leap year or not.

```
#include<stdio.h>
main()
{
int y;
printf("\n enter any year=");
scanf("%d",&y);
if( (y%4==0) && (y%100!=0) || (y%400==0) )
printf("\n %d is leap year",y);
else
printf("\n %d is not a leap year",y);
}
```

### **Assignments based on if and if else**

1. Write a program to accept an integer and check if it is even or odd.
2. Accept a character as input and check whether the character is a digit. (Check if it is in the range '0' to '9' both inclusive)
3. Write a program, which accepts annual basic salary of an employee and calculates and displays the Income tax as per the following rules.

Basic: < 1,50,000                      Tax = 0

Basic: 1,50,000 to 3,00,000 Tax = 20%

Basic: > 3,00,000                      Tax = 30%

4. Accept a character from the user and check whether the character is a vowel or consonant. (Hint: a,e,i,o,u, A, E, I, O, U are vowels)
5. Accept any year as input through the keyboard. Write a program to check whether the year is a leap year or not. (Hint leap year is divisible by 4 and not by 100 or divisible by 400)
6. Write a program to check whether given character is a digit or a character in lowercase or uppercase alphabet. (Hint ASCII value of digit is between 48 to 58 and Lowercase characters have ASCII values in the range of 97 to 122, uppercase is between 65 and 90)
7. Accept the x and y coordinate of a point and find the quadrant in which the point lies.
8. Accept the cost price and selling price from the keyboard. Find out if the seller has made a profit or loss and display how much profit or loss has been made.

Signature of the instructor

Date

## Exercise 4: Decision Making using Switch

### ❖ The switch statement:

C has a built-in multiple branch decision statement called switch. Switch is multiway decision making statement.

Syntax:

```
switch(expression)
{
    case value_1: statement part_1;
        break;
    case value_2: statement part_2;
        break;
    case value_n: statement part_n;
        break;
    default: default statement part;
}
```

### Assignments based on switch control structure:

1. Accept a single digit from the user and display it in words. For example, if digit entered is 9, display Nine.
2. Write a program, which accepts two integers and an operator as a character (+ - \* /), performs the corresponding operation and displays the result.
3. Accept two numbers in variables x and y from the user and perform the following operations

Option	Actions
1. Equality	Check if x is equal to y
2. Less Than	Check if x is less than y
3. Quotient and Remainder	Divide x by y and display the quotient and remainder

4. Accept radius from the user and write a program having menu with the following options and corresponding actions

Options	Actions
1. Area of Circle	Compute area of circle and print
2. Circumference of Circle	Compute Circumference of circle and print
3. Volume of Sphere	Compute Volume of Sphere and print

5. Write a program having a menu with the following options and corresponding actions

Options	Actions
---------	---------

1. Area of square	Accept length ,Compute area of square and print
2. Area of Rectangle	Accept length and breadth, Compute area of rectangle and print
3. Area of triangle	Accept base and height , Compute area of triangle and print

Signature of the instructor

Date

 /  /

## Exercise 5: Loop Control structures

Start Date

/	/
---	---

### ❖ Loop:

A segment of program code that is executed repeatedly is called a loop. The repetition is done until some condition for termination of the loop is satisfied.

Loops can be of two types:

- 1) top tested loop (entry controlled loop)
- 2) Bottom tested loop (exit controlled loop)

In top tested loop, the condition is evaluated before the loop body is executed. In the bottom tested loop, the condition is tested after the loop body is executed.

The C language provides three loop structures

- 1) While loop
- 2) Do ..while loop
- 3) For loop

### 1. The while loop:

The while loop is top tested loop i.e. the condition is tested before the loop body is executed.

Syntax:

```
while (expression)
{
    Instructions;
}
```

The statement can be a single or compound statement. The expression is evaluated and statement is executed as long as the expression is true. As soon as the expression becomes false, the execution of the loop body is stopped and control is transferred to the first statement outside the loop body.

Following program illustrate while loop

```
#include<stdio.h>
main()
{
    int i=1;
    while(i<=5)
    {
        printf("\n %d",i);
        i++;
    }
}
```

### 2. The do..while loop:

It is bottom tested loop or exit controlled loop i.e. it evaluates the condition after executing the statements. This means that the statement within the loop is executed at least once.

Following program illustrate do..while loop

```
#include<stdio.h>
main()
```



```

{
int i=1;
do
{
printf("\n %d",i);
i++;
}while(i<=5);
}

```

### 3. The for loop:

This is a top tested loop similar to the while loop but the advantage is that it combines the initialization test condition and loop variable alteration statement in a single statement.

Syntax;

```

for (initialization; condition; increment/decrement)
{
    Instructions;
}

```

Following program illustrate for loop

```

#include<stdio.h>
main()
{
int i;
for( i=0;i<=10;i++)
printf("\n%d",i);
}

```

### Assignments based on Loop Control Structure

- 1) Write a program to calculate sum of digits of a given input number.
- 2) Write a program to accept two numbers as a range and display sum of all numbers between that range.
- 3) Write program to check whether a input number is Armstrong number or not.
- 4) Write a program to accept a binary number and convert it into decimal number.
- 5) Write a program to check whether a input number is perfect number or not.
- 6) Write a program to calculate  $x^y$ .
- 7) Write a program to check whether a input number is palindrome or not.
- 8) Write a program to display multiplication of two input numbers without using \* operator.
- 9) Write a program to calculate sum of first and last digit of a number.

- 10)** Write a program to accept a number and count number of even, odd, zero digits within that number.
- 11)** Write a program, which accepts a number n and displays each digit in words. Example: 6702 Output = Six-Seven-Zero-Two. (Hint: Reverse the number and use a switch statement)

Signature of the instructor

Date

## Exercise 6: Nested Loops

### ❖ Nested loop:

Nesting of loops means a loop that is contained within another loop. Any loop can be nested within any other loop. The nesting is done such that the inner loop should be completely enclosed in the outer loop.

Eg.

While(expression)	for ( )	do
{	{	{
for( )	for( )	while( )
{	{	{
}	}	}
}	}	}while();

### Assignment based on Nested Loop

- 1) Write a program to generate following triangle up to n lines.

```
1
1 2
1 2 3
```

- 2) Write a program to generate following triangle up to n lines.

```
1
2 3
4 5 6
```

- 3) Write a program to generate following triangle up to n lines.

```
A B C
D E
F
```

- 4) Write a program to generate following pattern.

```
5
4 5
3 4 5
2 3 4 5
1 2 3 4 5
```

5) Write a program to generate following pattern.

Aa

Aa Bb

Aa Bb Cc

Aa Bb Cc Dd

- 6) Write a program to accept two numbers as range and display multiplication table of all numbers within that range.
- 7) Write a program to display all Armstrong between 1 and 1000.
- 8) Write a program to calculate sum of all digits of a input number till it reduce to a single digit.  
(Ex i/p 489 output sum:21 sum :3)

Signature of Instructor

Date

## Exercise 7: Demonstration of 1-D Arrays

The variable allows us to store a single value at a time, what if we want to store roll no. of 100 students? For this task, we have to declare 100 variables, then assign values to each of them. What if there are 10000 students or more? As you can see declaring that many variables for a single entity (i.e student) is not a good idea. In a situation like these arrays provide a better way to store data.

An array is a collection of one or more values of the same type. Each value is called an element of the array. The elements of the array share the same variable name but each element has its own unique index number (also known as a subscript). An array can be of any type, For example: int, float, char etc. If an array is of type int then it's elements must be of type int only.

Arrays can be single or multidimensional. The number of subscript or index determines the dimensions of the array. An array of one dimension is known as a one-dimensional array or 1-D array, while an array of two dimensions is known as a two-dimensional array or 2-D array.

### ❖ Declaring Arrays

To declare an array in C, a programmer specifies the type of the elements and the number of elements required by an array as follows –

```
type arrayName [ arraySize ];
```

This is called a *single-dimensional* array. The **arraySize** must be an integer constant greater than zero and **type** can be any valid C data type. For example, to declare a 10-element array called **balance** of type double, use this statement –

```
double balance[10];
```

Here *balance* is a array variable which is sufficient to hold up to 10 double numbers.

Index 0      1              2              3              4              5              6              7              8              9

--	--	--	--	--	--	--	--	--	--

balance[10]

All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.

### ❖ Initializing Arrays:

You can initialize an array in C either one by one or using a single statement as follows –

```
double balance[5] = {1000.0, 2.0, 3.4, 7.0, 50.0};
```

The number of values between braces { } cannot be larger than the number of elements that we declare for the array between square brackets [ ].

If you omit the size of the array, an array just big enough to hold the initialization is created. Therefore, if you write –

```
double balance[] = {1000.0, 2.0, 3.4, 7.0, 50.0};
```

You will create exactly the same array as you did in the previous example. Following is an example to assign a single element of the array –

```
balance[4] = 50.0;
```

The above statement assigns the 5<sup>th</sup> element in the array with a value of 50.0. All arrays have 0 as the index of their first element which is also called the base index and the last index of an array will be total size of the array minus 1.

**Assignment based on 1D Array:**

- 1) Write a program to accept n elements of 1D array and then display sum of all elements of array.
- 2) Write a program to find maximum and minimum elements of 1D array.
- 3) Write a program to calculate sum of all odd elements of 1-D array.
- 4) Write a program to sort 1D array elements in ascending order.
- 5) Write a program to display union of two 1D array.
- 6) Write a program to display intersection of two 1D array.
- 7) Write a program to display 1D array elements in reverse order.
- 8) Write a program to accept a decimal number and convert it to binary, octal and hexadecimal number.
- 9) Write a program to count the occurrences of a input number in 1D array.
- 10) Write a program to count Prime Numbers within 1D array.

Signature of the Instructor

Date

## Exercise 8: Demonstration of 2-D Arrays

### ❖ Two-dimensional Arrays

The simplest form of multidimensional array is the two-dimensional array. A two-dimensional array is, in essence, a list of one-dimensional arrays. To declare a two-dimensional integer array of size  $[x][y]$ , you would write something as follows –

```
type arrayName [ x ][ y ];
```

Where **type** can be any valid C data type and **arrayName** will be a valid C identifier. A two-dimensional array can be considered as a table which will have x number of rows and y number of columns. A two-dimensional array **a**, which contains three rows and four columns can be shown as follows –

```
int a[3][4];
```

a[0][0]	a[0][1]	a[0][2]	a[0][3]
a[1][0]	a[1][1]	a[1][2]	a[1][3]
a[2][0]	a[2][1]	a[2][2]	a[2][3]

Thus, every element in the array **a** is identified by an element name of the form **a[ i ][ j ]**, where 'a' is the name of the array, and 'i' and 'j' are the row and column subscript respectively that uniquely identify each element in 'a'.

### ❖ Initializing Two-Dimensional Arrays

Multidimensional arrays may be initialized by specifying bracketed values for each row. Following is an array with 3 rows and each row has 4 columns.

```
int a[3][4] = {  
    {0, 1, 2, 3} , /* initializers for row indexed by 0 */  
    {4, 5, 6, 7} , /* initializers for row indexed by 1 */  
    {8, 9, 10, 11} /* initializers for row indexed by 2 */  
};
```

The nested braces, which indicate the intended row, are optional. The following initialization is equivalent to the previous example –

```
int a[3][4] = {0,1,2,3,4,5,6,7,8,9,10,11};
```

### Accessing Two-Dimensional Array Elements

An element in a two-dimensional array is accessed by using the subscripts, i.e., row index and column index of the array. For example –

```
int val = a[2][3];
```

The above statement will take the 4th element from the 3rd row of the array.

### Assignments based on 2D arrays:

- 1) Write a program to find maximum and minimum elements of a matrix.
- 2) Write a program to calculate sum of all elements of a matrix.
- 3) Write a program to calculate sum of all even elements of a matrix.
- 4) Write a program to calculate sum of all upper triangular matrix elements.
- 5) Write a program to calculate sum of all diagonal elements of a matrix.
- 6) Write a program to calculate addition of two matrices of order  $m \times n$ .
- 7) Write a program to calculate multiplication of two matrices.
- 8) Write a program to display transpose of a matrix.
- 9) Write a program to calculate sum of elements of each row and column.
- 10) Write a program to check whether a input matrix is identity matrix or not.

Signature of Instructor

Date



## Exercise 9: Strings

### ❖ Strings in C

Strings are defined as an array of characters. The difference between a character array and a string is the string is terminated with a special character ‘\0’.

**Declaration of strings:** Declaring a string is as simple as declaring a one dimensional array. Below is the basic syntax for declaring a string.

```
char str_name[size];
```

In the above syntax str\_name is any name given to the string variable and size is used to define the length of the string, i.e the number of characters string will store. Please keep in mind that there is an extra terminating character which is the Null character (‘\0’) used to indicate termination of string which differs strings from normal character arrays.

**Initializing a String:** A string can be initialized in different ways. We will explain this with the help of an example. Below is an example to declare a string with name as str and initialize it with “Pune”.

1. `char str[] = "Pune";`
2. `char str[50] = "Pune";`
3. `char str[] = {'P','u','n','e'};`

### String Input and Output:

Functions gets() and puts() are two string functions to take string input from the user and display it respectively. Following program illustrate string input and output

```
#include<stdio.h>

void main()
{
    char name[30];
    printf("Enter name: ");
    gets(name);    //Function to read string from user.
    printf("Name: ");
    puts(name);    //Function to display string.
}
```

### ❖ Standard String Handling Functions:

Sr.No.	Function & Purpose
1	<b>strcpy(s1, s2);</b> Copies string s2 into string s1.
2	<b>strcat(s1, s2);</b>

	Concatenates string s2 onto the end of string s1.
3	<b>strlen(s1);</b> Returns the length of string s1.
4	<b>strcmp(s1, s2);</b> Returns 0 if s1 and s2 are the same; less than 0 if s1<s2; greater than 0 if s1>s2.
5	<b>strchr(s1, ch);</b> Returns a pointer to the first occurrence of character ch in string s1.
6	<b>strstr(s1, s2);</b> Returns a pointer to the first occurrence of string s2 in string s1.

### Assignments based on Strings:

- Write a menu driven program to perform the following operations on strings using standard library functions:  
1. Length 2. Copy 3. Concatenation 4. Compare
- Write a program which accepts a sentence from the user and alters it as follows: Every space is replaced by \*, case of all alphabets is reversed, digits are replaced by ?
- Write a program to accept a string and then count the occurrences of a specific character of a string.
- Write a program to count the occurrences of vowel from a input string.
- Write a program to calculate length of string without using standard functions.
- Write a program to copy string without using standard function.
- Write a program to connect two different strings without using standard function.

## Exercise 10: Functions

### ❖ WHAT IS C FUNCTION?

A large C program is divided into basic building blocks called C function. C function contains set of instructions enclosed by “{ }” which performs specific operation in a C program. Actually, Collection of these functions creates a C program.

#### Note:

- Any C program contains at least one function.
- If a program contains only one function, it must be main().
- If a C program contains more than one function, then one (and only one) of these functions must be main(), because program execution always begins with main().
- There is no limit on the number of functions that might be present in a C program.
- Each function in a program is called in the sequence specified by the function calls in main().
- After each function has done its task, control returns to main(). When main() runs out of function calls, the program ends.

**C functions can be classified into two categories,**

1. **Library functions**
2. **User-defined functions**

**Library functions** are those functions which are already defined in C library, example printf(), scanf(), strcat() etc. You just need to include appropriate header files to use these functions. These are already declared and defined in C libraries.

A **User-defined functions** on the other hand, are those functions which are defined by the user at the time of writing program. These functions are made for code reusability and for saving time and space.

#### **Benefits of Using Functions**

It provides modularity to your program's structure.

It makes your code reusable. You just have to call the function by its name to use it, wherever required.

In case of large programs with thousands of code lines, debugging and editing becomes easier if you use functions.

It makes the program more readable and easy to understand.

## C FUNCTION DECLARATION, FUNCTION CALL AND FUNCTION DEFINITION:

- There are 3 aspects in each C function. They are,
  - Function declaration or prototype – This informs compiler about the function name, function parameters and return value's data type.
  - Function call – This calls the actual function
- Function definition – This contains all the statements to be executed.

C functions aspects	Syntax
• function definition	• $\text{Return\_type} \quad \text{function\_name}$ (arguments list) { Body of function; }
• function call	• $\text{function\_name}(\text{arguments list});$
• function declaration	• $\text{return\_type} \quad \text{function\_name}$ (argument list);

### Assignments based on Functions:

1. Write a function isEven, which accepts an integer as parameter and returns 1 if the number is even, and 0 otherwise. Use this function in main to accept n numbers and check if they are even or odd.
2. Write a program to calculate sum of digits of a input number using function.
3. Write a program to calculate  $x^y$  using user defined function.
4. Write a function which accepts one number. Function should return 1 if the number is Perfect No, otherwise function should return 0. Use this function in main to display whether a input number is perfect number or not.
5. Write a function, which accepts a character and integer n as parameter and displays the next n characters.
6. Write a function isPrime, which accepts an integer as parameter and returns 1 if the number is prime and 0 otherwise. Use this function in main to display the first 10 prime numbers.
7. Write a program to find factorial of a input number using user defined function.
8. Write a program to calculate sum of following series up to n terms.

$$\text{Sum} = X + X^2/2! + X^3/3! + \dots$$

Signature of Instructor

Date

## Exercise 11: Pointers and Structure

Pointers in C language is a variable that stores/points the address of another variable. A Pointer in C is used to allocate memory dynamically i.e. at run time. The pointer variable might be belonging to any of the data type such as int, float, char, double, short etc.

Pointer Syntax : `data_type *var_name;`

Example : `int *p; char *p;`

Where, \* is used to denote that “p” is pointer variable and not a normal variable.

### KEY POINTS TO REMEMBER ABOUT POINTERS IN C:

- Normal variable stores the value whereas pointer variable stores the address of the variable.
- Always C pointer is initialized to null, i.e. `int *p = null`.
- The value of null pointer is 0.
- & symbol is used to get the address of the variable.
- \* symbol is used to get the value of the variable that the pointer is pointing to.
- If a pointer in C is assigned to NULL, it means it is pointing to nothing.
- Two pointers can be subtracted to know how many elements are available between these two pointers.
- But, Pointer addition, multiplication, division are not allowed.
- The size of any pointer is 2 byte (for 16 bit compiler).

### Structures in C:

#### What is a structure?

A structure is a user defined data type in C/C++. A structure creates a data type that can be used to group items of possibly different types into a single type.

#### How to create a structure?

‘struct’ keyword is used to create a structure.

#### How to declare structure variables?

A structure variable can either be declared with structure declaration or as a separate declaration like basic types.

Example :

```
struct address
{
    char name[50];
    char street[100];
    char city[50];
    char state[20];
    int pin;
};
```

#### How to access structure elements?

Structure members are accessed using dot (.) operator.

Array allows to define type of variables that can hold several data items of the same kind. Similarly **structure** is another user defined data type available in C that allows to combine data items of different kinds.

Structures are used to represent a record. Suppose you want to keep track of your books in a library. You might want to track the following attributes about each book –

- Title
- Author
- Subject

**Assignments based on Pointers and Structures:**

1. Write a program to display the elements of an array containing n integers in the reverse order using a pointer to the array.
2. Accept n integers in array A. Pass this array and two counter variables to a function which will set the first counter to the total number of even values in the array and the other to the total number of odd values. Display these counts in main. (Hint: Pass the addresses of the counters to the function)
3. Accept n integers in an array. Copy only the non-zero elements to another array (allocated using dynamic memory allocation). Calculate the sum and average of non-zero elements.
4. Write a program to allocate memory dynamically for n integers such that the memory is initialized to 0. Accept the data from the user and find the range of the data elements.
5. Accept the number of rows (m) and columns (n) for a matrix and dynamically allocate memory for the matrix. Accept and display the matrix using pointers.
6. Create a structure employee (id, name, salary). Accept details of n employees and write a menu driven program to perform the following operations. Write separate functions for the different options
  1. Search by name
  2. Search by id
  3. Display all
7. Create a structure Book (Bno, Bname, Price). Accept details of n Books and write a menu driven program to perform the following operations options.
  - i. Display all Books having price > 500
  - ii. Display Book having maximum price

Signature of Instructor

Date

**Section II**  
**Principles of Programming**  
**Algorithm**  
**(DOS, Scratch)**

### ❖ **DOS Commands:**

1. DIR – Displays directory of files and directories stored on disk.
2. DATE – Displays and/or sets the system date.
3. CLS – Clears the screen.
4. MD – Creates a new directory.
5. CD – Displays the name of or changes the current directory.
6. COPY CON- Used to create new file.
7. TYPE- Used to view the content of file.
8. COPY – Copies and appends files.
9. DEL – Deletes files from disk.
10. MOVE – Moves one or more files to the location you specify. Can also be used to rename directories.
11. ECHO – Displays messages or turns on or off the display of commands in a batch file.
12. EDIT – Starts the MS-DOS editor, a text editor used to create and edit ASCII text files.
13. FIND – Finds and reports the location of a specific string of text characters in one or more files.
14. LABEL – Creates or changes or deletes a volume label for a disk.
15. PATH – Sets or displays directories that will be searched for programs not in the current directory.
16. RENAME – Changes the filename under which a file is stored.
17. RD – Removes a subdirectory.

### ❖ **Windows Accessories:**

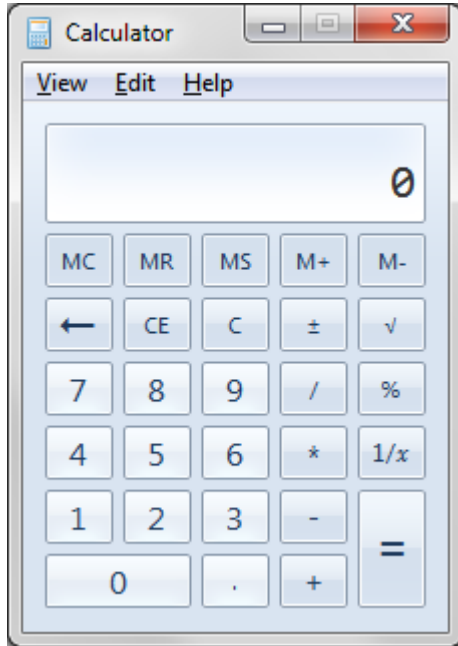
1. Calculator.
2. Snipping Tool.
3. Character Map.
4. Notepad.
5. WordPad.

Windows operating system ships with some handy applications known as Windows accessories. Calculator, Notepad, Paint, Explorer, WordPad is some of the most frequently used accessories.

Apart from above mentioned applications, Windows has a few tools for Ease of Access and some System Tools. We'll be briefly talking about them here.



## 1. Calculator:



Windows Calculator is a calculating application included in all the versions of Windows. It can be used to perform simple calculation, scientific calculation and Programming calculation.

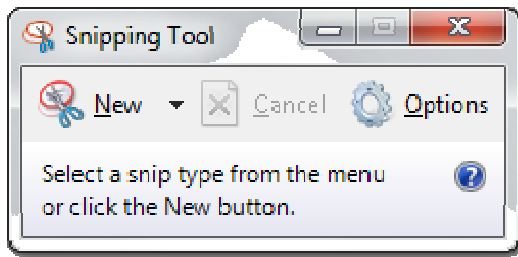
- Choose Start >> Programs >> Accessories >> Calculator to start Calculator application, or
- Alternately you can open Run dialog box (Start >> Run) dialog box then type *calc* and hit enter.
- From View menu choose the required type of calculator – Standard, Scientific, Programmer, Statistical
- Edit >> Copy and Paste commands can be used to input the numbers into calculator or paste the result to other applications

Calculator was first included with Windows 1.0 as a simple arithmetic calculator. In Windows 3.0, a scientific mode was added, which included exponents and roots, logarithms, factorial-based functions, trigonometry (supports radian, degree and gradians angles), base conversions (2, 8, 10, and 16), logic operations, Statistic functions such as single variable statistics and linear regression.

In Windows 7, Statistics, Unit Conversion, Date Calculation, and Worksheets modes were added. Calculator's interface was revamped for the first time since its introduction.

On the right of the main Calculator, one can add a panel with Date Calculation, Unit Conversion, and Worksheets. The included Worksheets allow one to calculate a result of a chosen field based on the values of other fields. Pre-defined templates include calculating a car's fuel economy (mpg and L/100 km), a vehicle lease, and a mortgage. In pre-beta versions of Windows 7, Calculator also provided a Wages template.

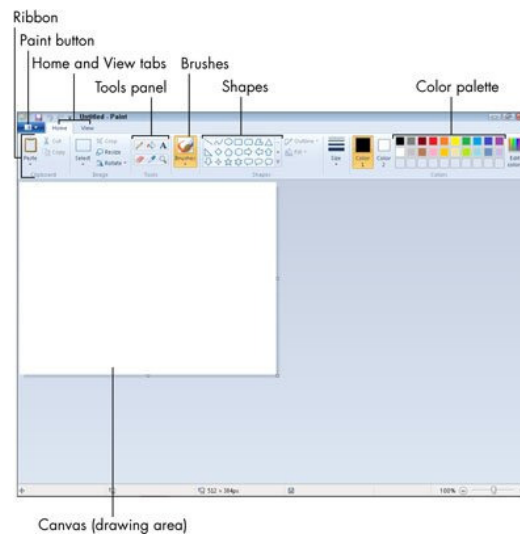
## 2. Snipping Tool:



Snipping Tool is another useful tool of Windows accessories included in Windows 7, Windows Vista, and Experience Pack for Windows XP Tablet PC Edition 2005. It was originally released as a power toy for the Tablet PC. The program is a screen-capture tool that allows taking screenshots (called snips) of an open window, rectangular areas, a free-form area, or the entire screen. Snips can then be annotated using a mouse or a tablet, saved as an image file (PNG, GIF, or JPEG file) or an HTML page, or e-mailed.

The Snipping Tool captures all or part of the computer display screen as a picture. You can save the picture and attach it to an e-mail or paste the picture into a document. Click the Start button, type snip, and click the Snipping Tool from the search results. The screen fades slightly, and the Snipping Tool toolbar appears. Snipping tool can be used to Print Screen, annotate and use in other applications.

## 3. Paint:

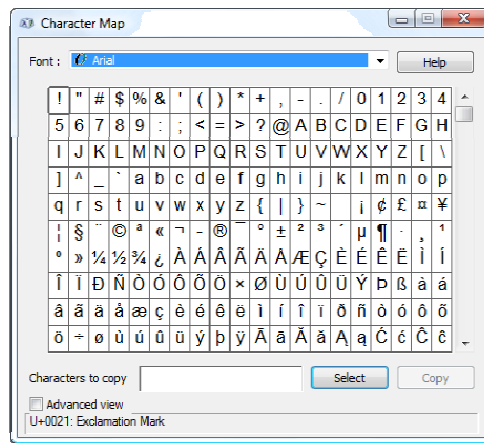


Paint (formerly Paintbrush for Windows) is a simple graphics painting program that has been included with all versions of Microsoft Windows. It is often referred to as MS Paint or Microsoft Paint. The program opens and saves files as

Windows bitmap (24-bit, 256 colors, 16 color, and monochrome) .BMP, JPEG, GIF. Paintbrush supports GIF without animation or transparency. Since Windows 98 Paint supports GIF with transparency. It also supports PNG (without alpha channel), and TIFF (without multiple page support).

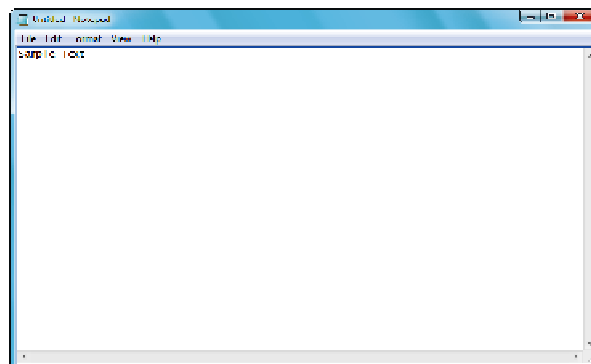
The program can be in color mode or two-color black-and-white, but there is no grayscale mode. For its simplicity, it rapidly became one of the most used applications in the early versions of Windows—introducing many to painting on a computer for the first time—and still has strong associations with the immediate usability of the old Windows workspace.

#### 4. Character Map:



Character Map is a utility included with Microsoft Windows operating systems and is used to view the characters in any installed font, to check what keyboard input (Alt code) is used to enter those characters, and to copy characters to the clipboard in lieu of typing them. The tool is usually useful for entering special characters. It can be opened via the command line or Run Command dialog using the 'char map' command.

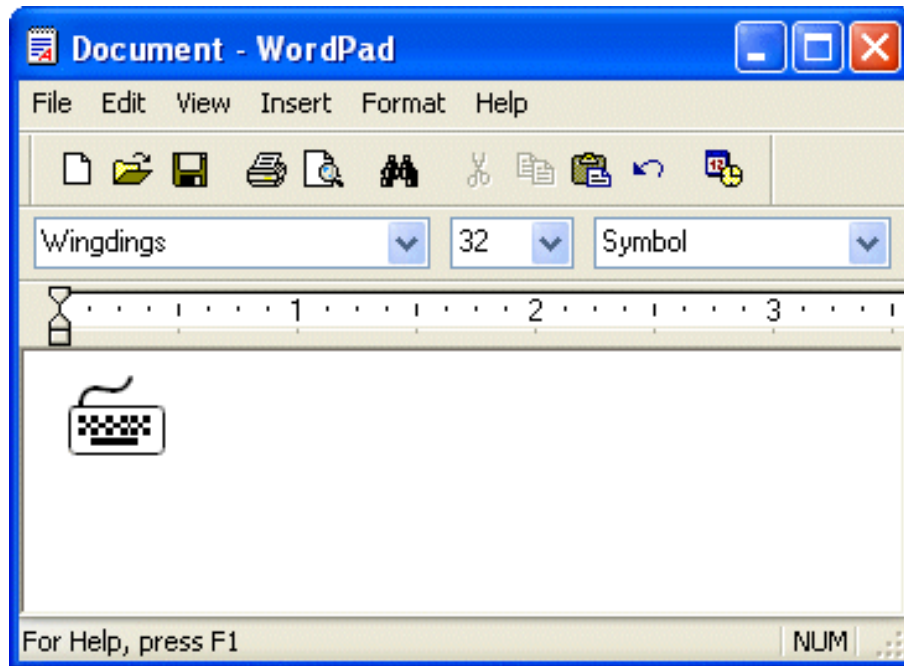
#### 5. Notepad:



Notepad is a common text-only (plain text) editor. The resulting files—typically saved with the .txt extension—have no format tags or styles, making the program suitable for editing system files that are to be used in a DOS environment

Notepad supports both left-to-right and right-to-left based languages, and one can alternate between these viewing formats by using the right or left Ctrl+Shift keys to go to right-to-left format or left-to-right format, respectively.

## 6. WordPad:



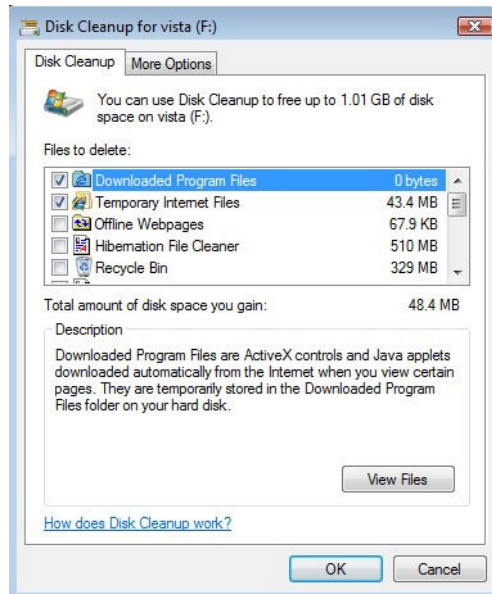
Microsoft WordPad is a free rich text editor included with Microsoft Windows 95 and the later. Earlier to Windows 95 there used to be an application called Write for the same task. Although capable of doing much more than Notepad, WordPad is not as advanced as Microsoft Word.

WordPad can format and print text, but lacks intermediate features such as a spell checker, thesaurus, and support for tables. As such, it is suitable for writing letters or short pieces, but underpowered for work that relies heavily on graphics or typesetting.

## ❖ System Tools:

For a smooth performance of a system, periodic maintenance is necessary. Tools like Disk Check, Disk Derangement, etc., helps in system maintenance. Windows includes some of the system utilities such as Disk Cleanup, Disk Defragment, System Restore, Control Panel and so on.

## ❖ Disk Cleanup:



In Windows 7, Vista, and XP, Disk Cleanup frees disk space by cleaning areas that gather unneeded files. Using Disk Cleanup regularly, along with ScanDisk, DEFRAG, and up-to-date virus patterns, will help keep your computer running smoothly.

### To run Disk Cleanup:

- From the Start menu, select Programs or All Programs, then Accessories, then System Tools, and then Disk Cleanup.
- Select the drive on which you would like to clear disk space (usually your C: drive), and click OK.
- Select from the four areas that Disk Cleanup will check and clean up:

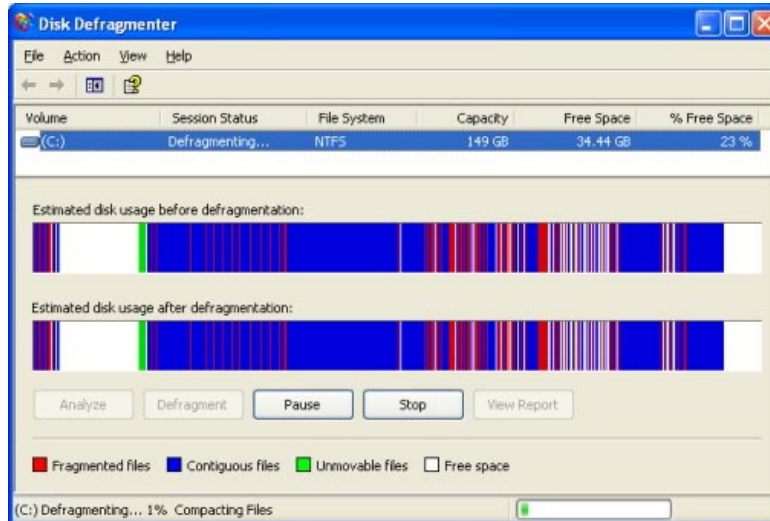
**Temporary Internet Files:** Files stored on your local drive that allow web pages to load more quickly.

**Downloaded Program Files:** A storage location for programs downloaded when you visit specific web sites that use ActiveX and Java applets.

**Recycle Bin:** Files that have been marked as deleted.

**Temporary Files:** Items in the TEMP folder.

## ❖ Disk Defragment:

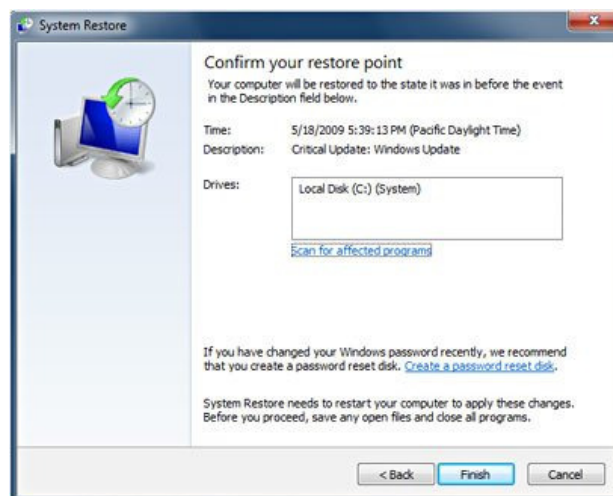


Disk Defragmenter consolidates files so that each is saved in contiguous physical space on the hard drive. When your computer crashes or freezes, the drive becomes fragmented. This means the files may be broken up and stored in different locations in your computer. This causes programs and files to run more slowly.

### To run Disk Defragmenter:

- Close all programs.
- Choose Start menu | All Programs | Accessories | System Tools | Disk Defragmenter.
- Select the drive you want to clean and click OK.
- Click Analyze. This may take several minutes.
- Click Close or Defragment. If the drive needs defragmented and you choose Defragment, this may take several minutes to several hours, depending on the number of files to consolidate.

### System Restore:

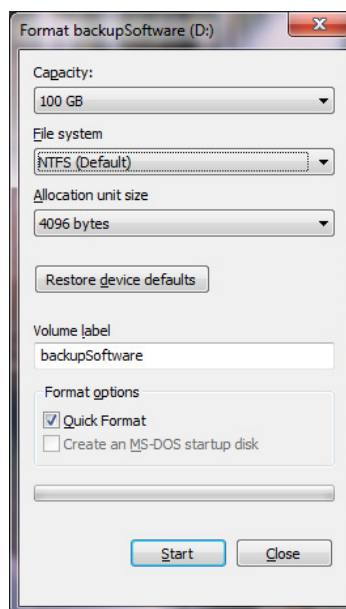


System restore is a feature of Windows that allows you to roll back the system to the same configuration it had at an earlier time. If this feature is turned on (which it is by default) Windows will take periodic snapshots (called restore points) of your installation. It can use the information gathered in these snapshots to restore your system to a previous configuration. It is a good idea to create a restore point before you make major changes to your system, such as before installing new hardware or doing any major updates.

### To run System Restore:

- Go to Start > Programs > Accessories > System Tools > System Restore.
- The System Restore screen will come up.
- Follow the on screen instructions

### Format:



There are times when Windows becomes so damaged or so compromised by hackers and Trojan programs that the only through solution is to essentially start all over. This is accomplished by formatting your hard drive, which means erasing all of the programs and the data on your machine, and re-installing Windows.

The format creates a new root directory and file system for the disk. It can also check for bad areas on the disk, and it can delete all data on the disk. To be able to use a new disk, you must first use this command to format the disk.

### ❖ Control Panel:

The **Control Panel** is a part of the Microsoft Windows graphical user interface. It allows users to view and manipulate basic system settings and controls via applets. You can use control panel for the tasks such as adding hardware, adding and removing software, controlling user accounts, and changing accessibility options. Additional applets can be provided by third party software.

In recent versions of Windows, the Control Panel has two views, Classic View and Category View, and it is possible to switch between these through an option that appears on either the left side or top of the window.

The classic view consists of shortcuts to the various control panel applets, usually without any description (other than the name). The categories are seen if the user uses “Details” view.

The category view consists of categories, which when clicked on display the control panel applets related to the category. In Windows Vista, the category used applets below the name of the category.

### **Microsoft Office:-**

#### **MS Word:**

- Introduction
- File Creations.
- File Operations
  - Save As, Open, Find and Replace, Cut, Copy, and Paste etc.
- Word Art
- Mail Merge
- Table
- Font
- Page Layout

#### **MS Excel:**

- Introduction
- Features of Excel
- Formula's
- Charts
- Working with multiple sheets.

#### **MS PowerPoint:**

- Introduction
- New slide insertion , Deletion
- Working with multiple slides
- Slide Show

#### **MS Access:**

- Introduction
- Database Creation
- Table creation
- Query
  - DDL, DML
- Form Creation
- Report Generation



### **Exercise: DOS and MS-Office**

1. Read and executes DOS commands given in this booklet.
2. Create your resume and send it to the multiple companies by using Mail Merge in MS Word.
3. Create mark sheet of student who is studying in FYBBA(CA).
4. Create MS PowerPoint Presentation for birthday party celebration.
5. Create a power point presentation for advertisement of College admission process.
6. Create a database DPU with two tables Teacher (tno,name,sal,desg,sub) and student(rollno,name,class,per). Establish one to many relationship between them with appropriate keys.
7. Design a form in MS Access for Email ID Registration with appropriate fields.
8. Generate a report of teacher who is having salary more than 15000 Rs. Assume Teacher table is already created.
9. Create sunrise view by using Paintbrush.

Signature of Instructor

Date:

# Scratch Programming

## 1. Assignments

1. Write a scratch script for dancing girls.
2. Write a scratch script for flying butterfly.
3. Write a scratch script for racing cars.
4. Write a scratch script for bouncing balls.
5. Write a scratch script for walking man.
6. Write a scratch script for blinking stars.
7. Write a scratch script for sunset.
8. Write a scratch script for sunrise.
9. Write a scratch script for birthday party celebration.
10. Write a scratch script for flag hoisting ceremony.
11. Write a scratch script for playing instrumental national anthem.

## 2. Mini Project using Scratch

Signature of Instructor

Date