

# Suggestive List of C++ Programs

1. Write a C++ program to display “Hello World!” on the output screen.
2. Write a program to display Multiplication Table of a number inputted by the user.
3. Write a program to read the coefficients A, B, C of the following quadratic equation and display the roots with appropriate message (nature of the roots).
$$A.X^2 + B.X + C = 0$$
4. Write a menu based program to compute the area and perimeter of the following geometrical figures:
  - (i) Circle
  - (ii) Square
  - (ii) Rectangle
5. Define the following user defined functions:  
  
double pow (double x ,int n) to calculate and return  $X^n$  ;  
double fact (int n) to calculate and return factorial of n ;  
  
Using these two defined functions, write programs to calculate the sum of the following series:
  - a. Sin Series
  - b. Cos Series
  - c. Exponential Series
6. Rewrite the above programs without using the functions pow( ) and fact( ).
7. Define a function to get an integer number N as argument and display all numbers of the Fibonacci Series equal or less than N. Also write a minimal C++ program to use this function.
8. Define a function SOD to get an integer number as argument and return the sum of the digits of that number. For example, if the argument is 1537, the function should return 16 (which is sum of 1 + 5 + 3 + 7). Also write a minimal C++ program to use this function.
9. WAP to read a string and determine whether that is a (non case sensitive) palindrome or not.  
Eg. Madam is a palindrome.  
(note ‘M’ & ‘m’ are equal in this non case sensitive example)

10. Write a program in C++ to read a sentence and calculate the following:
- (i) no. of lower case alphabets
  - (ii) no. of upper case alphabets
  - (iii) no. of special characters
  - (iv) no. of digits
  - (v) no. of words
11. WAP to read a string and encode as follow:
- The sequence of the words would be same but the characters of each word will appear in reverse order.
- Example:*
- |                         |                     |
|-------------------------|---------------------|
| <i>Original String:</i> | <b>I LOVE INDIA</b> |
| <i>Encoded String:</i>  | <b>I EVOL AIDNI</b> |
12. Write a program to read a name and then display the same but in sentence form as follow:
- If the inputted name is : MOHaNdas KaRAMChAnd GaNdhi
- The output would be : Mohandas Karamchand Gandhi
13. Define a function to read a name (string) as an argument and display the initials of that name with the full last name. e.g. for an argument 'MOHANDAS KARAMCHAND GANDHI', the output should be M K GANDHI.
14. Define a function HCF() to accept two arguments and return the HCF / GCD of those two numbers. Using this function, write a program to read two positive integers and then check whether they are co-prime or not. Two numbers are said to be co-prime when they don't have any common factor other than 1. (or HCF=1) e.g. 15 and 14 are co-prime.
15. Define a function ***isperfect()*** to check whether a number argument is a perfect number or not? Your function should return 1 if the number is perfect otherwise 0. Also write a minimal C++ program to use this function.
- Note:** A number is perfect if sum of all its factor is that number itself. Eg. 6 is a perfect no. as  $6=1+2+3$
16. Define a function to get a natural number as argument and return the same but in reverse order. For example, if the argument is 1537, the function should return 7351. Using this function, write a C++ program to check whether an inputted integer is palindrome or not.
17. Write a C++ program to display all prime factors of a number inputted by the user.
18. Define a function ***isprime()*** to read an integer argument and check whether that is a prime or not, if yes, the function returns 1 otherwise 0. Using this function display the first 10 prime members of the fibonacci series.

19. Define a function `SOF( )` to input an integer argument and then calculate and return the sum of all possible factors of that number argument (including 1 and excluding that number itself). Using this function WAP to read two numbers and check whether those numbers form an amicable pair or not. *(2 numbers are amicable when sum of factors of one number is equal to other and vice versa)*
20. Write a program to read a date and check for its validity.
21. Define and illustrate functions equivalent to the following pre-defined functions:  
`strlen( )`, `strcpy( )`, `strcat( )`, `isdigit( )`, `isupper( )`, `isalnum( )`
22. Write a program to display the following pattern when the number of maximum stars in a line is given: (When number of maximum stars in a line is 5, an odd number)
  - a)
 

```

A
AB
ABC
ABCD
ABCDE
          
```
  - b)
 

```

          1
        121
      12321
    1234321
  123454321
          
```
  - c)
 

```

                        *
                      * *
                    * * *
                  * * * *
                * * * * *
              * * * *
            * * *
          * *
        *
          
```
23. Declare a structure `COMPLEX` having two member variables `Real` and `Img`. Also define two ordinary functions to perform basic Complex algebra (`ADD`, `DIFFERENCE`). Write a menu based complete C++ program to use the above structure and functions.
24. Declare a structure `VECTOR` and define ordinary functions to perform basic VECTOR algebra (`ADD`, `DIFFERENCE`, `SCALAR MULTIPLICATION` and `VECTOR MULTIPLICATION`). Write a menu based complete C++ program to use the above structure and functions.
25. Declare a structure `BANK` having the members like name, balance, etc. Also write a menu based program to implement some basic operation (withdraw, deposit, etc. ) with this structure for 5 Account holders (objects).

26. Declare a structure STUDENT having the following members:

name (string),  
marks[5] (an array of integers)

Also write a menu based program to implement this structure for 10 students (objects)

**NOTE:** A student is passed only if  $\text{Total} \geq 200$  else "FAIL" and Total is the sum of all marks: marks[0]..marks[4]

27. Write a complete program to read the names of five political parties along with number of votes in favour of them. Display the name of the winning party. Use a structure having two member variables party and Votes.
28. WA menu based OOP to process the COMPLEX algebra (ADD, DIFFERENCE).
29. WA menu based OOP to process the VECTOR algebra (ADD, DIFFERENCE, SCALAR MULTIPLICATION and VECTOR MULTIPLICATION).
30. Declare a class EMPLOYEE having the following members:

**private** members:

**name**                      an array of 20 characters  
**basic**                     a float value

**public** members:

**getdata()**                a function to initialize the member variables  
**showdata()**             a function to display the member variables  
                              along with total salary (basic + DA)  
**DA()**                     a function to return 9% of basic

Declare an array of 10 objects of type EMPLOYEE.  
Initialize all Object using getdata() member function.

31. Declare a class LIBRARY having the following members:

**private** members:

**title**                      an array of 20 characters  
**author**                   an array of 20 characters  
**issuedto**                an integer number

**public** members:

**readdata()**                a function to initialize the member variables  
**displaydata()**           a function to display the member variables  
**issue()**                    a function to read membership number of the borrower  
                              and store into the variable issuedto  
**return()**                 a function to make issuedto variable empty (blank)

Declare an array of 10 objects of type LIBRARY.  
Initialize all Object using readdata() member function.

32. Declare and define a class BANK having the following members. Also write a menu based program to implement this class for 5 Account holders.
- ```
private:
    name, balance
    withdraw( )
    deposit( )
public:
    getdata( )
    showdata( )
    transaction( ) // to invoke withdraw( ) and deposit( ) depending upon user's choice
```
33. Declare and define a class STUDENT having the following members. Also write a menu based program to implement this class for 10 students.
- ```
private:
    name (string), marks[5] (an array of integers)
    result( ) // returns RESULT, which is "PASS" if Total >= 200 else "FAIL"
public:
    getdata( )
    showdata( )
```
- NOTE:** Total is the sum of all marks: marks[0]..marks[4]
34. Declare and define three functions having same name area( ) to calculate area of square, rectangle and triangle (using HERO's formulae). Write the complete menu based to use these functions effectively.
35. Define the following functions having same name PRIME( ) and also write the required program to demonstrate their use.
- If one argument is passed, the function should check whether the number is a prime and return 1 (if yes) or 0 (if not) accordingly.
  - If two argument is passed, then the function should check whether the numbers are co-prime or not and return 1 (if yes) or 0 (if not) accordingly.
36. Write a program to read an array of float 10 floating point numbers. Then calculate their Range. (Note: Range = Maximum – Minimum value stored in the array)
37. Define a function to rearrange an (argument) array of integers. All the negative numbers are shifted toward leftward whereas the positive members of the array are shifted toward right as shown below:
- |                  |   |    |    |    |    |   |    |    |
|------------------|---|----|----|----|----|---|----|----|
| Original Array   | : | 2  | -3 | -5 | 6  | 1 | -7 | -2 |
| Rearranged Array | : | -3 | -5 | -7 | -2 | 2 | 6  | 1  |

Write a program to read an array of integers and display the same after rearrangement using the defined function.

38. WAP to declare and read the marks of 4 students in 3 subjects (use 2-D array) and then calculate their total. Also calculate subject wise average marks scored by the students.
39. Write a complete program to read the names and salary in last twelve months by 100 employees. Finally display the names with their annual income.
40. WAP to declare and read the sales (in Rs.) made by 4 salesmen in 3 items (commodities) [use 2-D array] and then calculate the followings
- Total sale in Rs. for all salesmen
  - Item wise total sale.
  - Item wise maximum sale (for individual salesman)
  - Max Sale made by any salesman
41. Write a minimal program for Number Guessing game.
42. Write a program to check mathematical aptitude of a user. The program generates two three digit random numbers and asks the user to calculate and enter the sum. The program also checks the answer and display appropriate message.
43. WA Function to search whether an element float DATA is present in a sorted array float A[N] or not (use **Binary Search Technique**). Also write a minimum program to invoke the function.
44. WA Function to search whether an element int DATA is present in a sorted array int A[N] or not (use **Sequential / Linear Search Technique**). Also write a minimum program to invoke the function.
45. WA Function to search whether an element DATA is present in a sorted array A[N] or not (use **Binary Search Technique**). Also write a minimum program to invoke the function.
- when A[] is an array of following structure “student” :**
- ```
struct student  
{  
  char name[20];  
  int marks;  
};
```
- DATA is also an object of type student.**  
**Sorting is done on basis of marks of the students.**
46. WAF to accept an array float A[N] of float numbers and then return the same array but in ascending order. (Use **Sequential / Linear Sort Technique**). Also write a minimum C++ program to illustrate the defined function.

47. WAF to accept an array `int A[N]` of float numbers and then return the same array but in descending order. (Use **Bubble Sort Technique**). Also write a minimum C++ program to illustrate the defined function.

48. WA Function to search whether an element DATA is present in a sorted array `A[N]` or not (use **Sequential / Linear Search Technique**). Also write a minimum program to invoke the function.

when `A[]` is an array of following structure "student" :

```
struct student
```

```
{
```

```
char name[20];
```

```
int marks;
```

```
};
```

DATA is also an object of type student.

Sorting is done on basis of marks of the students.

49. WAF to accept an array `A[N]` of float numbers and then return the same array but in ascending order. (Use **Selection Sort Technique**). Also write a minimum C++ program to illustrate the defined function.

when `A[]` is an array of following structure "student" :

```
struct student
```

```
{
```

```
char name[20];
```

```
int marks;
```

```
};
```

Sorting to be processed on basis of marks of the students.

50. WAF to accept an array `A[N]` of float numbers and then return the same array but in descending order. (Use **Bubble Sort Technique**). Also write a minimum C++ program to illustrate the defined function.

when `A[]` is an array of following structure "student" :

```
struct student
```

```
{
```

```
char name[20];
```

```
int marks;
```

```
};
```

Sorting to be processed on basis of marks of the students.