

## **Basic Console IO**

1. Write a program to print “Welcome to C++” using escape sequences: \n, \t, \r, \. Observe the difference in the output.
2. Write a program to accept marks of five subjects from the user and calculate their average. Use implicit and explicit type conversion.
3. Write a program to add two numbers and store the result in a third variable. Print the result.
4. Write a program to swap two variables using a third variable and without using third variable.
5. Write a program to calculate Net Salary of an employee. Accept Basic Salary (BS) from the user.  
HRA is 15% of BS  
DA is 30% of BS  
PF is 12.5% of GS  
Gross Salary is BS + HRA + DA  
Net Salary = Gross Salary – PF
6. Accept a character from user. It may be alphabet, digit or any other character. Print its ASCII value.

## **Selection Statements**

7. Write a program to find leap year by using:
  - a. if-else and logical operators (&& and ||)
  - b. Conditional Operators (? :)(A leap year is divisible by 4 and is not divisible by 100 but it could be divisible by 400)
8. Write a program to accept the basic salary and total sales amount for a salesperson and calculate commission according to sales amount. Display the net salary and commission earned.  
(Net Salary = Basic Salary + Commission) The range is as follows.

Sales Amount in Rupees	Commission on Sales
5000 to 7500	3%
7501 to 10500	8%
10501 to 15000	11%
15000 and above	15%

9. Using switch-case construct, write a menu driven program to perform basic calculations (addition, subtraction, multiplication and division) on two user entered numbers.
10. Write a program to find maximum of three numbers using conditional operators.
11. Write a program to convert user entered character into its opposite case. Program should also flash an error message if the character entered by the user is not an alphabet.

## **Loops**

12. Write a program to print whether user entered number is an Armstrong number. Armstrong number is one for which the sum of the cube of all its digits is same as the number. For example,  $153 = (1*1*1) + (5*5*5) + (3*3*3)$
13. Write a program to display whether a user entered number is prime or not.
14. Draw a rectangle for user defined dimensions of length and breadth.
15. Write a program to generate all possible combinations of 1, 2, 3 using for loop.
16. Write a program for matchstick game between the computer and the user. Your program should ensure that the computer always wins. Rules for the game are as follows:
  - a. There are 21 matchsticks
  - b. The computer asks the player to pick 1, 2, 3, or 4 matchsticks.
  - c. Player is given the chance to pick the sticks first then the computer picks the sticks.
  - d. Whoever is forced to pick up the last matchstick loses the game.
17. Write a program to generate following output.
 

```

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

### **Functions – Call by Value**

18. Write a program to generate following table:

Data Type	Size in Bytes
Integer	4
Character	1
Float	4

The border of the above table should be generated using a function 'charLine()'. This function accepts a character and the length of the line to be printed, entered by the user in main(). Use the character to print the line.

19. Write a function 'Fibonacci()' to generate the terms of the fibonacci series. The number of terms should be entered by the user in main(). The series should be printed in the function. The terms of the Fibonacci series are 0 1 1 2 3 5 8 13 21 . . .
20. Find the sine value of an angle by calculating the following series up to first 5 items

$$\sin(x) = x - x^3/3! + x^5/5! - x^7/7! + x^9/9!$$

Where x is in radians.

Radian = 180/PI

PI = 3.142

Make use of following functions:

int factorial(int); and int power(int, int);

### **Functions – Call by Address**

21. Write a menu driven program, which allows the user to select either a circle or rectangle.
  - a. Write a single function areaCircum() that calculates the area and circumference of a circle. The values of area and circumference should be printed in main()
  - b. Write a single function areaPeri() that calculates the area and perimeter of a rectangle. The values of area and perimeter should be printed in main()
22. Write a program to accept date, month, year from the user in a function called getDate(), and print that date in main() in dd/mm/yy format.
23. Write a program which accepts two integers in main(). Pass these as arguments to function max(). In max(), find the maximum of these two numbers and using a return statement, return the address of the maximum number and print it in main().

### **Arrays**

24. Accept five integers in an array and use separate functions to:
  - a. Find maximum and minimum of the integers. Do not sort the array.
  - b. Multiply each element of the array by 5 and store it in another array and display it.
25. Write a program to delete an element from a user entered array. Accept position of element to be deleted from user.
26. Write a program to insert an element into an array. Accept position of element to be inserted from user.
27. Write a program to accept and display 3 by 3 matrix. Write Accept() and Display() functions to perform the tasks.

- a. Find the transpose of a matrix and print the transpose using display() function.
- b. Accept another matrix of same dimensions. Find the addition of two matrices and print the resultant matrix.

28. Write a program which will calculate multiplication of two 3 by 3 matrices.

29. Write a program to accept a string from the user. Write separate functions to perform the following:

- a. Find the number of occurrences of given character in the string
- b. Find the number of blank spaces in the string.
- c. Find the number of words in the string.
- d. Find the total number of all the vowels in the string.

### **Dynamic Memory Allocation**

30. Write a program that calculates the average marks of all the subjects. The number of subjects 'n' is accepted from the user. Allocate memory dynamically for 'n' integers. Free the memory when not in use.

31. Write a program to sort five user entered strings into alphabetical order. Use dynamic memory allocation to store strings.

### **Structures**

32. Define a structure "Book" having members – bookId, title, price. Use typedef to name this structure "BOOK". Accept the data for a book and display the record.

33. Modify above assignment - 32 to hold record of five books. Display the records of all the books using a function.

34. Write a program to copy one structure into another –

- a. On element by element basis.
- b. Copying an entire structure to another.

### **C++ Console IO**

35. Write a program to accept marks of a student in 'n' subjects. Allocate the space for the array of marks dynamically. Display average of marks. Use cin, cout, new, delete.
36. Implement function "Swap" to swap two integers. Use call by reference.
37. Overload function "Add" to add two integers, floats and doubles.
38. Write a single function "Multiply" to multiply two or three or four integers passed depending on call. Use default value arguments.

### **Classes and Objects**

39. Implement class Complex with real (int) and imaginary (int) parts as data members of class. Implement below functions inside this class.  
default constructor, parameterized constructor, getter functions for real and imaginary parts, setter functions for real and imaginary parts, display function.

40. Write a program to count number of objects created for above class Complex . Use static data member and member function.

41. Write a class Array which will store integer elements in dynamically allocated space.

```
class Array
{
    int size;
    int *arr;
}
```

Provide below functions in Array class -

1. Default constructor - Take 5 as default size. Initialize the array elements to 0
2. Parameterized constructor - Takes size as parameter. Initialize the array elements to 0.
3. Copy constructor - Performs deep copy of Array object.
4. Destructor - Deletes memory which is allocated dynamically.

42. Write a class Employee for an application which will have data members for employee id, employee name and salary. Provide the following functionalities in Employee class.

1. Initializing objects using default and parameterized constructors.
2. Accepting and displaying the information of employee from console

```
class Employee
{
```

```

        int emp_id;
        char emp_name[20];
        double salary;
    public:
        Employee();
        Employee(int, char*, double);
        void Accept();
        void Display();
};

```

In "main" function declare an array of Employee objects. Write a menu driven program to -

- a. Insert record into an array.
- b. Update information of specific employee on the basis of emp\_id accepted from user
- c. Display all records.

### **Operator Overloading**

43. Create objects c1,c2,c3,c4 of class Complex implemented earlier. Overload necessary operators to execute following statements

- a.  $c3=c1+c2$
- b.  $c3=c1-c2$
- c.  $c2=-c1$
- d.  $c4=++c1$
- e.  $c4=c1++$

44. Modify class Array in assignment – 41 to overload below operators

1. Overloaded [] operator - used to get/set array element e.g. arr[4] gives 4<sup>th</sup> index element
2. Overloaded ! operator - reverses array elements.
3. Overload >> - Accepts array size from user. Also accepts elements of array from user.
4. Overload << - Displays array.

45. Write a class Matrix which will store elements of integer matrix (two dimensional array) in dynamically allocated space. Provide below functions in matrix class -

1. Default constructor - Take 3 as default row and column size. Initialize the matrix elements to 0
2. Parameterized constructor - Takes row and column size as parameters. Initialize matrix to 0.
3. Copy constructor - Performs deep copy of matrix object.
4. Destructor - Deletes memory which is allocated dynamically.
5. Overloaded + operator - Performs addition of two matrices.
6. Overloaded ! operator - Calculates transpose of a user entered square matrix.
7. Overload >> - Accepts row & column size from user. Also accepts matrix elements from user.

8. Overload << - Displays matrix.

### **Containment and Inheritance**

46. Write a class Student having following –

- a. Student Roll Number (int)
- b. Student Name (char array)
- c. Date of Birth (Date class object where Date is user defined class)

Implement default constructor, parameterized constructor, accept, display. Generate the student roll number automatically.

47. Construct a hierarchy of employees.

1. Create an Employee class with attributes like employee id, name, date of birth.
2. Inherit class WageEmployee from base class Employee
3. WageEmployee class should have following members
  - a. Number of hours worked
  - b. Rate per hour
4. SalesPerson class should have following members
  - a. Number of items sold
  - b. Commission per item
5. Write constructors for WageEmployee and SalesPerson classes.
6. Override the functions for displaying details, calculating salary in WageEmployee and SalesPerson class.  
WageEmployee Salary = hours \* rate  
SalesPerson Salary = hours\*rate + sales\*commission

### **Polymorphism**

48. Modify above assignment 47 as given below –

- a. Create Employee pointer and allow it to hold address of any Employee objects (WageEmployee & SalesPerson) created on heap
- b. Using Employee pointer call accept() and display() to accept and display the details of dynamically created objects.
- c. Free the memory for the objects created on heap
- d. Make the Employee class as abstract.

### **Multiple Inheritance**

49. Extend the assignment 47 as per below requirements.
- Derive class Manager publicly from class Employee with data members for basic salary(int) and number of sub-ordinates(int).
  - Implement necessary functions inside class Manager
  - Derive class SalesManager publicly from SalesPerson and Manager. SalesManager do not have any of its own data members.
  - Implements necessary functions inside class SalesManager.

### **Templates**

50. Write a template function “Sort” to sort data of any kind. e. g. int, char, float, double

51. Write a template class Stack (LIFO) having -

Date members – Size(int), Top of stack (int), Pointer of type T

Functionalities- Default constructor, push, pop, display

Test this code for pushing integers, doubles, complex objects.

### **Exception Handling**

52. Modify above class Stack in assignment 50 to handle “overflow” and “underflow” exceptions using exception handling feature.

53. Create a class Account with -

Instance variable - balance

Class functions - deposit/withdraw

User withdrawal limit on one transaction is Rs.15000/-

Throw and Handle Exceptions -

1. OverLimit - when user tries to withdraw more than Rs. 15000/- in a transaction

2. InsufficientBalance - When user withdrawal amount is more than existing balance

### **STL**

54. Write a program using STL vector. The program should read in ten integers from the standard input and store each value in the vector. Now perform below operations.

- Insert two more values into the vector at the end of the vector.
- Print the vector using subscript notation.
- Remove two elements of vector from end.
- Print the vector using iterator.