Chapter 2:Basics of C++

1. Write a program to display “Hello World”
2. Write a program to calculate sum of 2 numbers.
3. Write a program to get user’s personal info & display it
4. write a program to demonstrate use of special sequence in c++.

Chapter 3:Input/Output in C++

1. Write a program to demonstrate use of Implicit and Explicit Type casting.
2. Write a program to demonstrate use of cerr & clog.
3. Create user-defined header file and use that header file in your main program.

Chapter 4:Operators & Control statements

1. Find the sum of 3 numbers.
2. Find the minimum number from 3 numbers
3. Display the grade according to given percentage(marks) of student.
4. Write a program to demonstrate used of bitwise operator.
5. Write a program to demonstrate used of Assignment operator.
6. Find the simple interest of given P,R,N
7. Write a program to demonstrate used of Nested switch.
8. Find the whether entered character is uppercase, lowercase , digit or special character using switch case statement.
9. Write a program to convert all -ve numbers of array into +ve number using if statement. ** like input : {1,-34,-4,3,6,0} => output :{1,34,4,3,6,0} */*
10. Write a program to which print all even number between given range[start : end].
11. Write a program to display the table between given range[start to end].

Chapter 5 : the collection of data

1. Write a program which takes student marks and display invalid marks. */* NOTE : invalid marks means marks couldn’t be less than 0 and greater than 100.*/*
2. Write a program to get sum of 2 matrix in following pattern.

```
+---+       +---+       +---+
| 256 |     | 121 |     | 377 |
| 452 |  +  | 236 |  =  | 588 |
| 321 |     | 613 |     | 934 |
+---+       +---+       +---+
```

3. Write a program which display following matrix.

```
+ – – — – +
| 1 0 0 0 0 |
| 0 1 0 0 0 |
| 0 0 1 0 0 |
| 0 0 0 1 0 |
| 0 0 0 0 1 |
+ – – — – +
```

4. Write a program which sort the no of word in ascending order. Use string class for it.
5. Create 5th dimension array and display on console.
6. Write a program which sort the array in descending order.
7. Write a program to find how many zeroes, -ve and +ve are present in the array.

Chapter 6: Oop's concepts

1. Create a class for collecting student data & display on console.
2. Write a program which demonstrate constructor overloading
3. write a program which takes no of employee data and display merit according to their salary.
4. Write a program that demonstrate the use of multilevel inheritance.
5. Write a program that demonstrate the use of multiple inheritance.
6. Create the hierarchy of following classes and access with child class.
   "Student → result." use result class and display data of all student.
7. Write a program which demonstrate the polymorphism for sum.
8. Write a program to demonstrate run time polymorphism.

Chapter 7: Functions – The group of Statements

1. Write a program which perform all arithmetic operation on given numbers and operator. *Use switch case to perform specific operation and write function for specific arithmetic operation.*
2. Create a user-defined header file which have one class with two methods which are used to find factorial and Fibonacci series respectively.
3. Write a program to swap two numbers using call by reference.
4. Write a recursion function to display all prime numbers within given range.
5. Write a recursion function which display following pattern where n = 4.

```
*
* * *
* * * * *
* * * * * * * * *
```

Chapter 8: Structure and Union

1. Write a program which collect the student data with help of Student Structure.
2. Define nested structure ‘Result{maths, science, english,total,percentage,status}’ of main structure of ‘Student{rollno,name,div,Result}’. Collect & calculate all fields from user and display it. */* NOTE : here **Result.status** is a boolean variable which represents that student is passed or failed based on all subjects marks Those curly brackets represents members of the structure*/*
3. Write a program which takes number of student’s data as object array and display all data with respect to student’s merit. For merit use user-defined function.

Chapter 9: Pointers

1. Display address of the any 3 variables which have different data types.
2. Create a program which sort the array using array to pointer. */* int a[100], int *ptr = a*/*
3. Create a pointer of an object and demonstrate with any example.

Character 10: File management

1. Write a program to store some data into file named ‘helloWorld.md’
2. Write a program to copy one file into another file.
3. Write a program to store ascii values and its symbols of a given range into a file.
4. Write a program to display the content of file in reverse.
5. Write a program to hide a message in a image or video.
6. Write a program which takes basic info Book*{title,author,edition,year}* class and store address of an object in append able file called ‘BookSell.txt’.
7. Create a file named emp.txt collect employee data and store into that file. Use Employee *{empno, empName,empSalary, empJoinDate}* Union.

Character 11: String the class

1. Write a program to display address of all character stored in char array.
2. Write a program to perform Caesar cipher algorithm in user message.
3. Write a program to convert string into toggle case. */*for Example : HeLLo_Me → hEllO_mE*/*
4. Write a program to take user name and password and check validation of user name and password which are stored in a file. */*Hint : create a file and store user name and password manually and then through program access them and check whether they are valid user or not.*/*
5. Write a program to find whether a given substring in present in a string or not.

Character 12: Templates

1. Create template for a any function to perform any operation */*NOTE : create valid example for this Task.*/*
2. Create template for class to calculate sum of 2 numbers with any 2 different data type./*hint use Template <class T,class M>*/*

Chapter 13: Exception Handling

1. Raise an Exception when user entered -ve Number.

2. Raise 2 exceptions first one for invalid user name and password, second one for invalid subject name. */*for 1st Exception set valid user name and password in the normal strings & for 2nd Exception set valid number of subjects in string array , like sub[] = {"Maths", "Science", "English"}*/*
3. Raise an Exception when data not found in the array.
4. Raise an Exception when given file name is not found.
5. Create user-defined Exception class and throw it's Object.

Chapter 14: Pre-processor directives

1. Create user-defined header file, in it create a Student class to get basic info of student & use this header file in another program.
2. Create a program to check whether a macro define or not.
3. Define a macro to calculate percentage of 3 subjects marks.
4. Create a program which checks balance is valid or not . */*Use #if */*