Chapter-1

1. Write a program to convert temperature from Fahrenheit to Celsius or Celsius to Fahrenheit based on user choice.
2. Write a program to swap value of two variables without using third variable. Don’t use operators + and -.
3. Time is given in seconds. Write a program to convert it into Hours, Minutes and Seconds. For example, if input is 6784 seconds then output is 01:53:04.

Chapter-2

4. Write a program to check whether entered number is even or odd without using modulus operator.
5. Write a program to check whether two numbers are equal or not without using relational operators.
6. Write a program to find out area of circle, area of triangle or area of square, area of rectangle based on user choice using switch statement.

Chapter-3

7. Write a program to print Pascal’s triangle up to n rows.
8. Write a program to print Floyd’s triangle up to n rows.
9. An electricity board charges the following rates to domestic users to discourage large consumption of energy:

For the first 100 units	- 60P per unit
For next 200 units	- 80P per unit
Beyond 300 units	- 90P per unit

All users are charged a minimum of Rs 50.00. If the total amount is more than Rs. 300.00 then an additional surcharge of 15% is added.

Write a program to read the names of users and number of units consumed and print out the charges with names.

Chapter-4

10. Write a program that obtains the largest of three numbers using inline function. Test the function using a main program.
11. Write a function power() to raise a number m to a power n. the function takes a double value for m and int value for n, and returns the result correctly. Use a default value of 2 for n to make the function to calculate squares when this argument is omitted. Write a main that gets the values of m and n from the user to test the function.
12. Write a program using the function overloading concept to find out the area of different geometrical figures like – circles, triangles, squares, and rectangles based on user choice.
13. Write a program to swap value of two variables passed to a function using call by reference.
14. Write a program to find out sum of two numbers using command-line arguments concept.

Chapter-5

15. Define a class to represent a bank account. Include the following members:

    Data members

    1. Name of the depositor
    2. Account number
    3. Type of account
    4. Balance amount in the account

    Member functions

    1. To assign initial values
    2. To deposit an amount
    3. To withdraw an amount after checking the balance
    4. To display name and balance

    Write a main program to test the program.

16. Write a class to represent a vector (a series of float values). Include member functions to perform the following tasks:

    (a) To create the vector
    (b) To modify the value of a given element
    (c) To multiply by a scalar value
    (d) To display the vector in the form (10,20,30, …)

    Write a program to test your class.

17. Create two classes DM and DB which store the value of distances. DM stores distances in meters and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB.

    Use a friend function to carry out the addition operation. The object that stores the results may be a DM object or DB object, depending on the units in which the results are required.

    The display should be in the format of feet and inches or meters and centimeters depending on the object on display.
18. Write a menu-driven program to add, subtract and multiply two complex objects when passed to a member function.
19. Write a menu-driven program to add and subtract two time objects when passed to a member function.
20. Write a menu-driven program to add and subtract two distance objects when passed to a member function.
21. Write a menu-driven program to add, and compare two string objects when passed to a member function.
22. Write a menu-driven program to add, subtract and multiply two complex objects when passed to a friend function.
23. Write a menu-driven program to add, subtract and multiply two matrices objects when passed to a friend function.
24. Write a program to show the friend function as a bridge between two non-interlinked classes and swap their data-member using that friend function.
25. Write a program to implement the friend class. Design a class Number having two instance variables- num1 and num2 and a member function to initialize these instance variables. Now make class Operations, a friend class to class Number and perform arithmetic operations on private instance variables of class Number.

26. Write a program to implement the nested class. Design a class student having instance variables- name, rollno and a nested class Date with read() and write(0 functions to read and print date respectively. Create dob, doj and doe objects of Date class inside student class and add two member functions to class student read() and write() to read and print instance variables of name, rollno and also the date of birth, date of joining and date of enrollment.
27. Write a program to implement the local class. Design a method having local class to illustrate the features and rules of local class. Design class for student inside function.
28. Program to calculate the average percentage of marks in a particular subject by n students of a class using static members and static member functions.
29. Design a Stack class and implement basic stack operations.
30. Design Queue class and implement basic queue operations.

Chapter-6

31. Define a class String that could work as a user-defined string type. Include constructors that will enable us to create an uninitialized string

    String s1; //string with length 0

    And also to initialize an object with a string constant at the time of creation like

    String s2("Well done!");

    Include a function that adds two strings to make a third string. Note that the statement

        s2=s1;

    will be perfectly reasonable expression to copy one string to another.

    Write a complete program to test your class to see that it does the following tasks:

    (a) Creates uninitialized string objects.
    (b) Creates objects with string constants.
    (c) Concatenates two strings properly.
    (d) Displays a desired string object.
    (e) Destructor to free memory.

    Note: All constructors must be dynamic constructors and also include copy constructor and destructor.

32. A book shop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher and stock position. Whenever a customer wants a book, the sales person inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then the system displays the book details and requests for the number of copies required. If the requested copies are available, the total cost of the requested copies is displayed; otherwise the message "Required copies not in stock" is displayed.

    Design a system using a class called books with suitable member functions and constructors. Use new operator in constructors to allocate memory space required.

Chapter-7

33. Create a class FLOAT that contains one float data member. Overload all the four arithmetic operators so that they operate on the objects of FLOAT.
34. Create a class MAT of size m * n. Define all possible matrix operations for MAT type objects by overloading operators.
35. Define a class String. Define all possible operations on String type objects.
36. Write a program to overload insertion << and extraction >> operators.

37. Write a program to overload the assignment operator.
38. Write a program to overload the function call operator ().
39. Write a program to overload the array subscript operator [].
40. Write a program to overload class member access operator ->.
41. Write a program to overload the unary increment operator ++ and show pre-increment and post-increment operators are different when used in an expression.
42. Write a program to convert int type to Time type, where Time is a class having hours and minutes as instance variables.
43. Write a program to convert Time type int type, where Time is a class having hours and minutes as instance variables.
44. Write a program to convert Dollar type to Rupees type, where Dollar is a class having the dollar as an instance variable and Rupees is a class having rupees as an instance variable using the casting operator function.

Chapter-8

45. Assume that a bank maintains two kinds of accounts for customers, one called a savings account and the other as a current account. Both accounts provide deposit and withdrawal facilities. Maximum withdrawal at a time in case of a saving account is not more than 10,000 Rs. and in the current account, it is 50,000 Rs. The account holders should also maintain a minimum balance (5,000 Rs. in the case of a savings account and 10,000 Rs. in the case of a current account) and if the balance falls below this level, a service charge is imposed. The service charge in the case of the saving account is 500 Rs. but in the case of the current account, it is 1000 Rs.

    Create a class account that stores customer name, account number, balance and type of account. From this device the classes cur_acct and sav_acct to make them more specific to their requirements. Include necessary member functions in order to achieve the following tasks:

    (a) Accept deposit from a customer and update the balance.
    (b) Display the balance.
    (c) Compute and deposit interest.
    (d) Permit withdrawal and update the balance.
    (e) Check for the minimum balance, impose penalty, necessary, and update the balance.

    Note: Use constructors for initialization of instance variables.

46. An educational institution wishes to maintain a database of its employees. The database is divided into a number of classes whose hierarchical relationships are shown in the following figure. The figure also shows the minimum information required for each class. Specify all the classes and define functions to create the database and retrieve individual information as and when required.

    Class Staff having instance variables – code and name. It has three child classes – teacher, typist, and officer. The teacher class has instance variables- subject and publication. The typist class has an instance variable- speed. The officer class has an instance variable- grade. The typist class has two child classes- regular and casual. The regular class have instance variable- salary and casual class have instance variable- daily wages. Design main() method for implementation.

    Note: Use constructors for initialization of instance variables.

47. Implement multiple inheritance by deriving a class result from two parent classes- theory and practical. Use constructors for initialization of instance variables. Use initialization list also. Show how ambiguity can be resolved.
48. Consider a class network following figure. The class master derives information from both account and admin classes which in turn derive information from the class person. Define all the four classes and write a program to create, update and display the information contained in master objects. **(Multipath Inheritance).**

    The person class has instance variables- name and code. The account has an instance variable- pay. The admin class has instance variables- experience.

    Note: Use constructors for initialization of instance variables.

Chapter-9

49. Write a program to illustrate the concept of an array of pointers to objects. Design a class city having instance variables- name and len, where name is a character type pointer and len is int type variable. Now, define an array of pointers to objects so that each object's memory allocation is as per requirement that is not same for all objects.
50. Write a program to illustrate the concept of pointers to functions by performing addition and subtraction of two numbers passed to add and subtract functions respectively.
51. Design a Person class having two instance variables- name and age. Write a program to find the elder person out of two persons by designing an elder member function. Pass the object (second person) to this function and return the object who is elder from the elder function. Use this pointer inside the elder function to return the called object.
52. Consider a book shop that sells both books and video-tapes. Design a class media and derive two classes- book and tape from media class. The class media have instance variables- title and price of a publication. The class book has an instance variable playtime. A function display() is defined in each class to display the contents of the class. The display() function is defined as a virtual function in the media class.
53. Create an abstract base class called shape. Use this class to store two double-type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add to the base class, a member function get_data() to initialize base class data members and another member function display_area() to compute and display the area of figures. Make display_area() as a pure virtual function and redefine this function in the derived classes to suit their requirements.

    Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively, and display the area.

    Remember the two values given as input will be treated as lengths of two sides in the case of rectangles and as base and height in the case of triangles, and used as follows:

    Area of rectangle = x * y

    Area of triangle = ½ * x * y

Chapter-10

54. Write a program to read a list containing the item name, item code, and cost interactively and produce a three-column output as shown below.

Name	Code	Cost
Turbo C++	1001	250.95
C Primer	905	95.70
… … … …	… … … …	… … … …

    Note that the name and code are left-justified and the cost is right-justified with a precision of two digits. Trailing zeros are shown.

55. Write a program to design two manipulators- currency and form. The currency manipulator display Rs and the form manipulator sets -trailing zeros, positive sign for positive numbers, precision upto 2 places, width of 10 and use * to fill vacant places.
56. Write a program that reads a text from the keyboard and displays the following information on the screen in two columns:
    (a) Number of lines
    (b) Number of words
    (c) Number of characters
    Strings should be left-justified and numbers should be right-justified in a suitable field width.

Chapter-11

57. Write a program that reads a text file and creates another file that is identical except that every sequence of consecutive blank spaces is replaced by a single space.
58. A file contains a list of telephone numbers in the following form
    John 23456
    Ahmed 9876
    …………… …………
    The names contain only one word and the names and telephone numbers are separated by white spaces. Write a program to read this file and output the list in two columns. The names should be left justified and the numbers should be right justified.
59. Design a program to implement Random access file with operations – inserting record, deleting existing record, updating existing record, retrieving particular record, and display all records. Implement by taking the example of an employee.

Chapter-12

60. Write a function template for finding the minimum value contained in an array.
61. Write a function template for finding the roots of a quadratic equation.
62. Write a function template for sorting an array using Bubble sort.
63. Write a program to overload the function template.
64. Write a program to illustrate the concept of generic function restrictions.
65. Design a Stack template class and implement basic stack operations.
66. Design a Queue template class and implement basic queue operations.
67. Write a class template to represent a generic vector. Include member functions to perform the following tasks:
    (a) To create the vector
    (b) To modify the value of a given element
    (c) To multiply by a scalar value
    (d) To display the vector in the following form (10, 20, 30 …)

Chapter-13

68. Write a program with the following:
    (a) A function to read two double-type numbers from the keyboard.
    (b) A function to calculate the division of these two numbers.
    (c) A try block to throw an exception when a wrong type of data is keyed in.
    (d) A try block to detect and throw an exception if the condition "divide by zero" occurs
    (e) Appropriate catch block to handle the exceptions thrown
69. Write a program to design and implement user-defined exception. Design a function to find the square root of passed parameter but when parameter value is below 0.0001 then pass user-defined exception with message "number is too small".
70. Write a program to design and implement multiple-catch statements. Design a function to throw different types of exceptions based on int value passed to function. When 1 is passed then throw int exception, for 2 pass char exception, for 3 pass double exception, for 4 pass float exception, and in all other cases print message "value accepted".
71. Write a program to design and implement re-throwing an exception. Take example of division of two numbers passed to a function.
72. Write a program to design and implement catch all block (**catch(…)**). Take example of design a function test which receive an int type parameter and throw int, float, char and double type of exceptions based on parameter value as 1,2,3, and 4 respectively. Design catch all block to catch all exceptions.