

List of Practicals

1. Imagine a publishing company that markets both book and audiocassette versions of its works. Create a class publication that stores the title (a string) and price (type float)of a publication. From this class derive two classes: book, which adds a page count (type int), and tape, which adds a playing time in minutes (type float). Each of these three classes should have a getdata() function to get its data from the user at the keyboard, and a putdata() function to display its data.
Write a main() program to test the book and tape classes by creating instances of them, asking the user to fill in data with getdata() , and then displaying the data with putdata().
2. Write a class representing bank accounts. The class should have the following data members : Customer Name, Account Number, Type of Account (Savings/Current etc), Account Balance. The class should allow basic operations like creating a new account, deposit an amount, withdraw money after checking the balance, display account details.
3. (CarbonFootprint Abstract Class: Polymorphism) Using an abstract class with only pure virtual functions, you can specify similar behaviour for possibly disparate classes. Governments and companies worldwide are becoming increasingly concerned with carbon footprints (annual releases of carbon dioxide into the atmosphere) from buildings burning various types of fuels for heat, vehicles burning fuels for power, and the like. Many scientists blame these greenhouse gases for the phenomenon called global warming. Create three small classes unrelated by inheritance—classes Building, Car and Bicycle . Give each class some unique appropriate attributes and behaviours that it does not have in common with other classes. Write an abstract class CarbonFootprint with only a pure virtual getCarbonFootprint method. Have each of your classes inherit from that abstract class and implement the getCarbonFootprint method to calculate an appropriate carbon footprint for that class (check out a few websites that explain how to calculate carbon footprints). Write an application that creates objects of each of the three classes, places pointers to those objects in a vector of CarbonFootprint pointers, then iterates through the vector, polymorphically invoking each object's getCarbonFootprint method. For each object, print some identifying information and the object's carbon footprint.
4. Implement a resource allocator using the concept of static data member and static member function. A Resource class should have a static data member to keep track of resource allocation (Boolean variable). Instances of this class will be allocated resources only if no other instance is holding the resource.
5. Develop a simple linked list class, LinkedList, to hold a list of integers. A linked list consists of an arbitrary number of nodes which encapsulate both a data value and a pointer to the next node in the list. Implement a node as a Node class nested within LinkedList. Provide a LinkedList:: Add () member function which allows integers to be added to a LinkedList object.
6. We want to store information about different vehicles. Create a class named Vehicle with two data members named mileage and price. Create its two subclasses Car with data members to store ownership cost, warranty (by years), seating capacity and fuel type (diesel or petrol). Bike with data members to store the number of cylinders, number of gears, cooling type(air, liquid or oil), wheel type(alloys or spokes) and fuel tank size(in inches) Make another two subclasses Audi and Ford of Car, each having a data member to store the model type. Next, make two subclasses Bajaj and TVS, each having a data member to store the make-type.

Now, store and print the information of an Audi and a Ford car (i.e. model type, ownership cost, warranty, seating capacity, fuel type, mileage and price.) Do the same for Bajaj and TVS bike.

7. Write a Program to find Maximum out of Two Numbers using friend function.
Here one number is a member of one class and the other number is a member of some other class.
8. Write a Program to swap private data members of classes named as class_1, class_2 using friend function.
9. Write a Program to design a class to represent a matrix. The class should have the functionality to insert and retrieve the elements of the matrix. Define constructors to allocate memory dynamically for the dimensions supplied.
10. Write a Program which creates & uses an array of the object of a class. (for eg. implementing the list of students of a class having details such as Name, Age, marks etc..).
11. Write a class with a simple integer as a class member and a member function that returns that value. There should be two overloaded constructors one which takes an integer as an argument and the other taking a "char *" as an argument. (Note: Can use atoi function to convert "char *" to int.)
12. Overload the '+' operator to add two Rational Numbers.
13. Overload the '*' operator to multiply two matrices.
14. Design a **Point** class with coordinates x and y representing a point on the plane (use constructors and overload << and >> operators). Design a **Polar** class which represent class using polar coordinates radius(r) and angle(a). Write a program which takes input data for a Point object and perform **type conversion** so as to convert the Point data into Polar data.
Use the following formulas where required:
 $x = r * \cos(a)$; $y = r * \sin(a)$;
 $a = \text{atan}(y/x)$; $r = \text{sqrt}(x*x + y*y)$;
15. Define an Employee class with class members as ID, name and salary. Define a member function "greater" which takes one parameter (object of class Employee) as a function argument. The function should compare the salary of two Employees and return the object whose salary is greater. Achieve the above using "**this**" pointer.
16. Write a program to implement I/O operations on characters. I/O operations includes inputting a string, Calculating the length of the string, Storing the string in a file, fetching the stored characters from it, etc.
17. Write a program to copy the contents of one file to another.
18. Write a program to maintain an elementary database of employees using files.
19. Accept 3 file names as command-line arguments. The first 2 are input files in which first file has to be created as a float file and the second file has to be created as a string file. Merge the contents of these 2 files into the 3rd file. It should be one integer from the first file followed by one line from the second file. Display the merged file.
Design a user-defined manipulator to display the output for float values as
10 column width
Right-Justified
Two-digit precision
Filling unused space with '*'
Trailing zeros are shown
Show sign for positive values

20. Passing by reference is also an effective way to allow a function to return more than one value. Write a program that returns the previous and next power of two of the parameter passed using the concept of pass by reference.
21. Write a program to sort a list of numbers using call by reference. Smallest number should be output first.
22. Write a program in C++ that instantiates a function template that implements a binary search of a sorted array of objects.
23. Write a function to reverse the elements of an array. The function should be a template which can operate on any type of data. Invoke the reverse function for arrays with different data types from main().
24. Create a class Stack storing stack of integers. Use exception handling to deal with situation of overflow and underflow
25. Write a program to compute the square root of a number. If the number is negative, a user defined exception my_sqrt() should be raised