

## OOPS LAB EXPERIMENTS

1. Write a C and C++ program to read and print the details of 'N' students using array of structures in the c program and array of objects for the c++ program.

### Class and Object

1. Write a C++ program to find the factorial of a given number and check whether it is prime or not using class
2. Write a C++ program to find the prime numbers.
3. Write a C++ program to find the reverse of the string
4. Write a C++ program to print the length of a string using recursion
5. Create a class account that contains Customer name, account number, type of account, and balance. Implement member functions for the following operations
  1. To read the data members
  2. Accept deposit from customer
  3. Permit withdrawal
  4. Compute and display interest if no of years is given.
6. Create a class student with data members Name, Roll no, marks, branch, total. Implement member functions to read and display student details.

### Inline functions

1. Write a C++ program to find the square and cube of a number using inline function
2. Write a C++ program to find the largest among three numbers using inline function.

### Function overloading

1. Write a C++ program to find the area of triangle, rectangle and circle using the concept of function overloading
2. Write a C++ program to swap two variables using call by value and call by references.

### Static member and static member functions

1. Write a C++ program to count the number of objects of a class.
2. Write a C++ program to search for element in an array using static member function. Create appropriate Class and data members.

### Friend function

1. Write a C++ program to find sum of two complex numbers
2. Write a C++ program to find the smallest and largest among N numbers.
3. Write a C++ program to add the objects of two different classes using friend functions

### Constructor and Destructor

1. Create a class box with length, breadth and width as data members. Use default, parameterized and copy constructors to initialize the values for three objects and destructor to destroy the objects after use. Find the volume of these objects and display the results using appropriate member functions.

2. Create a class Date with data members date, month, year and string day. Use appropriate constructor to initialize the object and destructor to destroy. Display the date in the format "dd-mm-yyyy , day" using a member function.
3. Write a C++ program to find the area of a wall using copy constructor.

#### Operator overloading

1. Create a class Date with data members int date, int month, int year. Create a Date object and overload unary operator --(decrement) to find the previous Date.
2. Write a C++ program to add two boxes using operator overloading.
3. Create a class Distance with data members feet and inches. Overload binary + and < operators to add and compare two distance objects respectively. < operator function should be implemented as a friend function.
4. Write a program to overload stream operators << and >> to read and display the objects of complex class.

#### Inheritance

1. **Employee** class contains details like name, emp no, pay rate, constructor function and a pay() function. **Manager** class inherits from employee and has the option of drawing pay on hourly basis or salary basis and has an additional data is\_salaried(bool). Class **Supervisor** is derived from employee and has an additional field department and is always salaried. Base and both derived classes should contain pay() function with same name.
2. Create a class named Vehicle with two data member 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) Create objects of each class. Read and display all the data related to each object. Use appropriate member functions in each class.
3. Write a C++ program to implement the following classes and relations. Define functions appropriately. Display function should display results of all operations performed by member functions.

#### Pointers and virtual functions

1. Create a class book. Members are title and price. Derive two classes, textbooks and novels. Text books has members char class (I-V) and string subject name. Novels has data members string genre and int pages. Use member functions to read and display data in each class. Use the concept of pointers to objects to access data members and member functions.
2. Write a C++ program to create a class STUDENT with age name and register number. Using Inheritance, derive two classes MTech-stud and BTech-stud. List both the category of students in the increasing order of marks(for BTech-stud) and gpa( for MTech-stud). In case of tie, display whichever name comes first. Make sort() function as a virtual function.
3. Implement the base class Shape and derive triangle, rectangle, circle and square classes from it. Implement functions to compute the area and perimeter of the polygon. Use the concept of pure virtual functions.