**ASSIGNMENT**

**Subject: Object Oriented Programming in C++**

**Thinking Object Oriented**

1. Describe OOP as a new paradigm in your own words with a suitable example.
2. Why OOP dominates POP? Mention the characteristics of OOP.
3. Write short notes on:

a. Coping with complexity

b. Computation as simulation

c. Non-linear behavior of complexity.

**Class & Methods**

1. Explain various access specifiers (or access mode) used in C++ with an example.
2. What is a function? Discuss the use of friend function taking in to consideration the concept of data hiding in object oriented programming.
3. What is the difference between defining a member function inside and outside the class? Give suitable example.
4. Write a C++ program to calculate the factorial of a number using recursion.
5. Write a program to add two complex numbers using the concept of friend function.
6. Write short notes on:

a. Default argument

b. Inline function

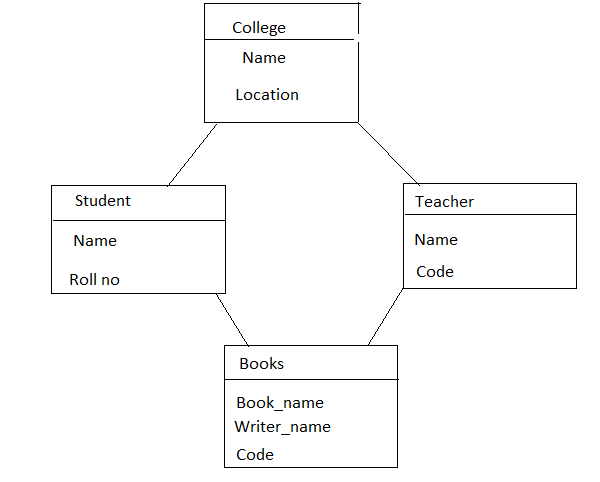
c. Static member function

**Message, Instance & Initialization**

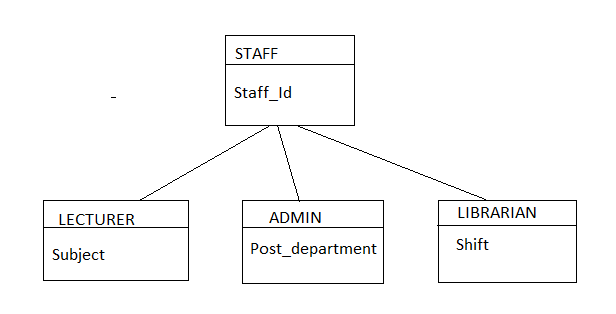
1. Explain the purpose of constructor and destructor. Describe their significances. Can we have more than one destructor in a class? Explain it.
2. WAP using constructor to find the area of a square and rectangle.
3. WAP to add two complex numbers using the concept of constructor.
4. What are advantages of copy constructor? Explain with example.
5. WAP to check whether a given number is Prime or Not, using Constructor.
6. What do you mean by Stack versus Heap Storage? Explain about memory recovery.
7. What is the difference between message passing and function call? Explain the basic message passing formalism.
8. What is constructor? Is it mandatory to use constructor in a class? Explain.

**Object Inheritance & Reusability**

1. What does inheritance mean in C++? Describe the syntax of the single and multiple inheritances. When do we use the protected visibility specifier to a class member?
2. What is hybrid inheritance? Does ambiguity occurs in this type of inheritance? If yes, explain.
3. State principle of substitutability. Explain sub classing for specialization, generalization. List out the disadvantages of inheritance.
4. A cassette shop deals with three items, namely Audio Cassettes, Video Cassettes and CD-ROMs. Define a class ITEM that can be used as base for all the above items, and provide suitable method to add and display item list for each of the derived class. The attributes can be Name, Serial Number and Company etc.
5. What are virtual base classes? Write a program to implement the concept of hybrid inheritance so as to find the result of the student (Consider there are four Classes: Student, test, Sports and Result and assume the necessary data as well as member functions).
6. What is an abstract class? How does it differ from a virtual base class? Explain.
7. How does OOP provide reusable software component? Explain.
8. How does inheritance reuse the existing code? Explain.
9. What is containership? How does it differ from inheritance?
10. Create an abstract base class shape with two members breadth and height, a member function for initialization and a pure virtual function to compute area(). Derive two specific classes Triangle and Rectangle which override the function area(). Use these classes in a main function and display the area of a triangle and a rectangle.
11. Compare and contrast IS-A rule and HAS-rule.
12. “Inheritance allows us to create a hierarchy of classes.” Justify this statement. Discuss the private and public inheritance.
13. Distinguish between a subclass and a subtype in a light of the principle of substitutability. Support your answer with examples.
14. The following figures shows minimum information required for each class. Write a program to realize the above program with necessary member functions to create the database and retrieve individual information. Every class should contain at least one constructor c and should be inherited to other classes as well.



1. Develop a complete program for an institution, which wishes to maintain a database of its staff. The database is divided into number of classes whose hierarchical relationship is shown in the following diagram. Specify all the classes and define constructors and function to create database and retrieve the individual information as per the requirement.



**Polymorphism**

1. What is virtual function? When do we make a function virtual and when we make a function pure virtual? Explain with suitable example.
2. What is operator overloading? Write a program to concatenate two strings using overloaded ‘+’ operator.
3. Write a program to implement vector addition and subtraction using operator overloading.
4. Explain polymorphic variables and deferred methods. When is the use of deferred method preferred?
5. What is polymorphism? Differentiate between compile time and run time polymorphism with program in each.(early binding and late binding)
6. Discuss the role of virtual functions in C++ to cause dynamic polymorphism. Show with example the how it is different from the compile time polymorphism.
7. How can you achieve the run-time polymorphism in C++? Explain.
8. Define two classes named 'polar' and 'rectangle' to represent points in polar and rectangle systems. Use type conversion to convert from one system to another system.

**Template and Generic programming**

1. What is generic programming? How would you convert regular classes into templates? Explain with example.
2. Write a program using template to add two integer; two floats and one integer and one float number respectively. Display the final result in float.
3. Define two classes named ‘Polar’ and ‘Rectangle’ to represent points in polar and rectangle systems. Use conversion routines to convert from one system to another system.
4. What is generic programming? How would you convert regular classes into templates? Explain with example.
5. Make a class called memory with member data to represent bytes, kilobytes and megabytes. Read the value of memory in bytes from user as basic type and display the result in user defined memory type. Like for m(basic type)= 108766, your program should display as: 1 megabyte 38 kilobytes 177 bytes. [Hint: use basic to user defined data conversion method].
6. Write a class template to represent a generic vector, include member functions to perform the following tasks.
7. To create a vector.
8. To modify the value of a given element.
9. To multiply by a scalar value.
10. To display the vector in the fo r (10, 20,30….)

**Object Oriented Design**

1. Differentiate between
2. Programming in small and programming in large.
3. Interface and Implementation.
4. What are the critical issues that are to be considered while designing the large programming? Why?
5. What are the advantages of adopting RDD (Responsibility Driven Design)? Explain with a help of a suitable example.
6. What is the purpose of CRC cards? Explain with example.
7. How is responsibility driven design (RDD) suitable to handle a complex problem? What do you mean by coupling and cohesion?
8. What are the basic steps involved in Responsibility Driven Design. How does it help in object oriented design? Support your answer with an example.
9. What is the role of behavior in OOP? Along with a figure and an example of CRC card, explain its significance in object oriented design.
10. Explain state, behavior and identity in support of object-oriented paradigm.
11. "Reusability implies non-interference." Justify the statement with suitable example.
12. What influence in an object oriented approach said to have on the software system design? Give your own opinion. Justify with an example.