

Lab Performance 03

1. A Student Database contains information of all the students of the University. A student's information consists of Student **Name**, Student **ID**, Student **Phone** and Student **CGPA**. Initially the number of Students of the University is 0. Whenever a new student joins the university the no of students is updated. There should be a mechanism to assign and read the information of each student. Also student information can be showed into the console.
 - a. Define the **behavior** of the class
 - b. The designed system should maintain **encapsulation**
 - c. All the relevant **set** and **get** methods should be designed
 - d. You must have 1 default **constructor**
 - e. You must have a **Parameterized Constructor** to assign initial values
 - f. You must use **destructor**
 - g. After finishing all the above mentioned tasks create **object(s)** and test the class

Decide whether you need **static member data or/and static member functions**.
2. Write a program that will implement a class which will use the following structure or blueprint:

| |
|--|
| Check |
| int a int b int c |
| 1 Default/ Empty Constructor() 1 Parameterized Constructor() Destructor() testNumber() ShowInfo() <i>[Hints: It will check whether two numbers are equal or not.]</i> |

3. Write a class **course** with the **course_title**, **course_code** and **credit** as **private** members of the class. Now make an **array** of 10 **course objects** in main function and demonstrate use of all methods (e.g. setData, getData etc.).

4. A **Company Management System(CMS)** needs to use information about *employees* having **salary, address, hiredate, phone, DOB** identified by **eno**, **departments** having **dname, location** and **budget** identified by **dno**. The company has multiple *branches* each having a *location*, a *managerID* and identified by **bno**. Employees work in departments. Each branch has multiple department. Employees who are working for **more than 12 months** are entitled salary bonus of 10%. This amount will be needed to be printed and not added to the employee's salary. You have been asked to **create the system** described above using **OOP principles** in a suitable programming language. In order to do that you will need to do the following:
- Find the **classes** and their **features**
 - Define the **behaviors** of **each** of the **classes**
 - When a and b are done, create **instances** of the classes to demonstrate their usage.
5. Create a class named "**AirShip**" which has following members:
- DATA MEMBERS:
- Private int passenger;
 - Private double cargo;
- MEMBER FUNCTIONS:
- Parameterized constructor that initializes the value of **passenger** and **cargo**.
 - A **show** method that prints **AirShip** information.

Create another Class named "**AirPlane**" which will inherit **AirShip** class using **protected inheritance**. The **AirPlane** includes the following members.

DATA MEMBERS:

- Private string engine; // example values : propeller , jet
- Private double range;

MEMBER FUNCTIONS:

- Parameterized constructor of AirPlane that initializes the value of **passenger, cargo, engine, range**
- A **show** method that prints **AirPlane** information.

6. Create a base class called *building* that stores the number of floors a building has the number of rooms and its total square footage. Create a derived class called *house* that inherits building and also stores the number of bedrooms and number of bathrooms. Next create a derived class called *office* that inherits building and also stores the number of fire extinguishers and the number of telephones.

7. Write a program which implements the following features:
- Write a class ***person*** which have following properties –
 - Two private variables named ***name*** and ***age***.
 - Two public functions named ***setName()*** and ***getName()*** which will **assign** value to ***name*** variable and will **return** that value respectively.
 - Two public functions named ***setAge()*** and ***getAge()*** which will **assign** value to ***age*** variable and will **return** that value respectively.
 - Write a class ***student*** which will inherit (public) class ***person***. And this class will also have its own following properties –
 - A private variable named ***cgpa***.
 - Two public functions named ***setCGPA()*** and ***getCGPA()*** which will **assign** value to ***cgpa*** variable and will **return** that value respectively.
 - Write a class ***teacher*** which will inherit (public) class ***person***. And this class will also have its own following properties –
 - A private variable named ***salary***.
 - Two public functions named ***setSalary()*** and ***getSalary()*** which will **assign** value to ***salary*** variable and will **return** that value respectively.

When **a**, **b** and **c** are done create ***instances*** of the classes to demonstrate their usage.

8. Write a program to declare a class with three data members. Declare ***overloaded constructors*** with ***no*** arguments, ***one*** argument, ***two*** arguments, and ***three*** arguments. Pass values in the object declaration statement.