Department of Computer Science and Engineering
RCC Institute of Information Technology
Canal South Road, Kolkata - 700015

Course Name: Object Oriented Programming Language Lab **Course Code: PCC-CS591**
Semester: 3rd Year, 5th Sem., B.Tech (CSE) **Academic Session: 2024-25**

Lab Assignment I

Fundamentals:

1. Write a Java Program to print your name.
2. Write a Java Program to calculate sum and average of two numbers (Use any data type). (Command Line Input).
3. Write a Java program to calculate a Factorial of a number. (using Scanner)
4. Write a Java program to calculate Fibonacci Series up to nth term (using Scanner)
5. Write a Java program to print addition of two matrix.
6. Write a Java Program to print all prime numbers within a given range.
7. Write a Java program to design a calculator using switch case.
8. Write a Java program to find the minimum in an array of integers.
9. Write a Java program to input integers in a two dimensional array and display the maximum value column wise.
10. Write a Java Program to convert a string to an integer in Java.
11. Write a Java program to implement Linear search.
12. Write a Java program to implement Selection sort.

Department of Computer Science and Engineering
RCC Institute of Information Technology
Canal South Road, Kolkata - 700015

Course Name: Object Oriented Programming Language Lab **Course Code: PCC-CS593**
Semester: 3rd Year, 5th Sem., B.Tech (CSE) **Academic Session: 2024-25**

Lab Assignment - II

Classes and Objects

1. Create a class Rectangle with two instance variables:
    a. length : int
    b. width : int

    and two methods:
    i. calcArea(): int
    ii. display(): void

    Use them from main method declared in the same class.

2. Create a class Stack and implement push() and pop() operations.

3. Create a class Complex and implement add(), subtract() and isequal() operation. (Use ‘this’ keyword)

4. Create a class Box with three instance variables:
    a. width : int
    b. height : int
    c. depth : int

    and three methods:
    i. volume(): double
    ii. display(): void
    iii. compareVolume(Box, Box): Box

    Create three instances of Box class and display the details of the object with highest volume.

5. Create a class Person having name and age of a Person as data members. Write a method that compare two persons by their age and print the person’s name and age who is senior most.

6. Design a Payroll class that has fields for an employee’s name, ID number, hourly pay rate and number of hours worked. Write the appropriate accessor and mutator methods and a constructor that accept the employee’s name and ID number as arguments. The class should also

have a method that returns the employee’s gross pay, which is calculated as the number of hours worked multiplied by the hourly pay rate. Write a program that demonstrates the class by creating a Payroll object, then asking the user to enter the data for an employee. The program should display the amount of gross pay earned.

7. Create a class Triangle with two instance variables:
    a. base: int
    b. height: int

    And two methods:

    i. area(): double
    ii. display(): void

   Use constructor to initialized the instance variables. Create an array of objects of type Triangle. Calculate and display the area of the object with highest area in main().

8. Write a program using parameterized constructors to calculate area of a square and a rectangle.

9. Create a class named Figure. Make cube, cylinder and sphere as object of the class Figure and calculate their surface area using concept of constructor overloading.

10. Design a TestScores class that has fields to hold three test scores. The class should have accessor and mutator methods for the test score fields, and a method that returns the average of the test scores. Demonstrate the class by writing a separate program that creates an instance of the class. The program should ask the user to enter three test scores, which are stored in the TestScores object. Then the program should display the average of scores, as reported by the TestScores object.

Lab Assignment - III

Inheritance

1. Write a class **College** with following data members:
   a) collegeName
   b) address
   and method:
   a)showDetails().
   Write a class **DepartmentFaculty** which inherits class **College** with following specialized data members:
   a) name
   b) experience
   and method:
   a) showFacultyDetails().
   Write a class **Department** which also inherits class **College** with following specialized data members:
   a) deptName
   b) hodName
   and method:
   a)showDeptDetails().
   Create object of all the classes and test your program from main().
   Add appropriate constructors, mutators and accessors wherever necessary.

2. **Employee** and **ProductionWorker** Classes: Design a class named Employee. The class should keep the following information in fields:
   i) Employee name
   ii) Employee number in the format XXX-L, where each X is a digit within the range 0-9 and the L is a letter within the range A-M.
   iii) Hire date.

   Write one or more constructors and the appropriate accessor and mutator methods for the class.
   Next, write a class named ProductionWorker that inherits from the Employee class. The ProductionWorker class should have fields to hold the following information:
   i) Shift (an integer)
   ii) Hourly pay rate (a double)

The workday is divided into two shifts: day and night. The shift will be an integer value representing the shift that the employee works. The day shift is shift 1 and the night shift is shift 2. Write one or more constructors and the appropriate accessor and mutator methods for the class. Demonstrate the classes by writing a program that uses a ProductionWorker object.

3. In a particular factory a shift supervisor is a salaried employee who supervises a shift. In addition to a salary, the shift supervisor earns a yearly bonus when his or her shift meets production goals. Design a **ShiftSupervisor** class that inherits from the **Employee** class. The ShiftSupervisor class should have a field that holds the annual salary and a field that holds the annual production bonus that a shift supervisor has earned. Write one or more constructors and the appropriate accessor and mutator methods for the class. Demonstrate the class by writing a program that uses a ShiftSupervisor object.

4. In a particular factory, a team leader is an hourly paid production worker who leads a small team. In addition to hourly pay, team leaders earn a fixed monthly bonus. Team leaders are required to attend a minimum number of hours of training per year. Design a TeamLeader class that inherits from the ProductionWorker class. The TeamLeader class should have fields for the monthly bonus amount, the required number of training hours, and the number of training hours that the team leader has attended. Write one or more constructors and appropriate accessor and mutator methods for the class. Demonstrate the class by writing a program that uses a TeamLeader object.

5. Ship , CruiseShip , and CargoShip Classes:
   Design a **Ship** class that the following members:
   - A field for the name of the ship (a string).
   - A field for the year that the ship was built (a string).
   - A constructor and appropriate accessors and mutators.
   - A toString method that displays the ship's name and the year it was built.

   Design a **CruiseShip** class that extends the Ship class. The CruiseShip class should have the following members:
   - A field for the maximum number of passengers (an int)
   - A constructor and appropriate accessors and mutators
   - A toString method that overrides the toString method in the base class.

   The **CruiseShip** class's toString method should display only the ship's name and the maximum number of passengers.
   Design a **CargoShip** class that extends the Ship class. The CargoShip class

should have the following members:

- A field for the cargo capacity in tonnage (an int)
- A constructor and appropriate accessors and mutators
- A toString method that overrides the toString method in the base class.

The CargoShip class’s toString method should display only the ship’s name and the ship’s cargo capacity.

Demonstrate the classes in a program that has a Ship array. Assign various Ship , CruiseShip , and CargoShip objects to the array elements. The program should then step through the array, calling each object’s toString method.

Lab Assignment - IV

Abstract and Interface

1. Design an interface named **Queue** with the following methods:
   a) To insert() and delete() elements from the Queue.
   b) display() to display the content of the modified Queue.

2. Create an abstract class **Accounts** with the following
   Data members:
   a. Balance
   b. AccountNo
   c. HoldersName
   Methods:
   a. withdraw()
   b. deposit()
   c. display()

   Create a subclass **SavingsAccount** of Accounts class and add the following details:
   Data member:
   a. rate_of_interest
   Methods:
   a. calculateAmount()
   b. display(), to show rate of interest along with account balance.

   Create another sub class **CurrentAccount** of Accounts class with the following details:
   Data member:
   a. overDraftLimit
   Method:
   a. display()- to show overdraft amount.

   Create object of these classes and call their methods.

2. Create an **class Student** with the following
   Data member:
   a. roll: int
   Methods:
   a. setRoll(int) //take input

b. getRoll(): int //display data member

Create a **subclass Test** and add the following details:

Data members:

a. marks of subject1 (sub1):double
b. marks of subject2 (sub2):double

Methods:

a. setMarks(double, double)
b. getMarks() to display marks of two subjects

Create **interface Attendance** with the following details:

Data member:

a. Atten: int

Method: getAttendance()

Create a **class Result** which extends Test class and implements **Attendance** with the following details:

Data member:

a. total: double

Methods:

a. getAttendance() //implements interface (display atten)
b. getTotal() //sub1+sub2+atten
c. display()

Create object of Result class and call the methods.

3. Create an abstract class **Shape** with the following
   Methods are: calculateArea():void
   Create a subclass **Circle** and add the following details:
   Data member:
   a. radius: int
   Method:
   a. calculateArea()
   Use appropriate construct to take a input

   Create object of Circle class and call the methods.

4. Create interface **Speaker** with method: speak(): void
   Create subclass **Lecturer** of Speaker with implement the interface Speaker.
   Create subclass **Politician** of Speaker with implement the interface Speaker.

   Create reference variable of Speaker and call the methods of all sub classes.

Department of Computer Science and Engineering
RCC Institute of Information Technology
Canal South Road, Kolkata - 700015

Course Name: Object Oriented Programming Language Lab **Course Code: PCC-CS 593**
Semester: 3rd Year, 5th Sem., B.Tech (CSE) **Academic Session: 2024-25**

Lab Assignment - 5

String

1. Write a Java program to input a sentence and display the words in sorted order.
2. Write a Java program to find the length of a String except white spaces.
3. Write a Java program to convert double value to String.
4. Write a Java program to check whether a String contains characters or digits or both.
5. Write a Java program to input two Strings and check whether the second String is the suffix of the first or vice-versa.
6. Write a Java program to compare two Strings and find if they are equal or not.
7. Write a Java program to check whether the letter ‘A’ or ‘a’ is present in a String or not.
8. Write a Java program to test if a String starts with a specified substring.
9. Write a Java program to remove “Information” from the String “RCC Institute of Information Technology”.
10. Write a Java program to take out 4th (lower bound) to 10th (upper bound) characters from a String.
11. Write a Java program to convert a String to lower case, upper case and toggle case.
12. Write a Java program to check for Palindrome String.
13. Write a Java program to insert a character/ substring at the beginning of a String.
14. Write a Java program to count number of occurrence of a character/ substring in a String.
15. Write a Java program to print the following the pattern:

a. Input: HELLO H HE HEL HELL HELLO	b. Input: HELLO H HE HEL HELL HELLO HELL HEL HE H

16. Write a Java program to input a sentence and display the word with highest frequency.

Department of Computer Science and Engineering
RCC Institute of Information Technology
Canal South Road, Kolkata - 700015

Course Name: Object Oriented Programming Language Lab **Course Code: PCC-CS 593**
Semester: 3rd Year, 5th Sem., B.Tech (CSE) **Academic Session: 2024-25**

Lab Assignment - 6

Package

1. Define two different classes namely, **Student** (**store in package stu**) and **Employee**. These classes are derived from a base class **Person** (**store in package per1**).
   Define other two classes **Staff** and **Faculty**. **Staff** and **Faculty** classes are derived from **Employee** class. **Employee**, **Staff** and **Faculty** are stored in the package **emp**.
   The **Person** class has name and age data and display method to display the name and age of a person.
   The **Student** class has data like rollNo and branch and display method to display name, age, rollNo and branch of the student.
   Faculty has designation data (Assistant Professor, Associate Professor and Professor) and display method to display the name, age, EmpId, doj and designation of the Faculty.
   Staff has EmpId and doj(date of joining) data, designation data (Technical and Clerical) and display method to display the name, age, EmpId, doj and designation of the Staff.
   Each class has their own constructor to initialize the value of each data field. Finally create **MainDemoClass** and create an object of each class. Print the values of all objects in the MainDemoClass. Store MainDemoClass in package **mainpkg1**. Create a package **collegeMgt** and store all the other packages under collegeMgt.

2. Create a package named **shapes**. Create classes Square, Triangle, Circle under the package **shapes.** The classes should contain mutator, accessor, calculateArea() and calculatePerimeter() methods. Create Main Class, store it in **mainpkg2** package and create an object of each class in main(). Store all the packages under package **geometry.**

3. Design a class named Person (store in package **per2**)with fields for holding a person’s name, address and telephone number. Write one or more constructors and the appropriate mutator and accessor methods for the class’ fields.

   Design another class named Customer (store in package **cust**), which inherits from the Person class.

   The Customer class should have a field for a customer number and a boolean field indicating whether the customer wishes to be on a mailing list. Write one or more constructors and the appropriate mutator and accessor methods for the class’s fields. Demonstrate an object of the Customer class in Main class (stores in the package **mainpkg3)**. Store all the packages under package **custDemo**.