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

Paper Name:Object Oriented Programming Lab **Paper code: CS594D**

Semester: 5th
Academic Session:

LIST OF LAB EXERCISES

DAY 1(BASIC CONCEPT OF JAVA ENVIRONMENT SETUP, SIMPLE PROGRAM WITHIN MAIN FUNCTION & BASIC CONCEPT OF CLASS):

1. Introduction to Java Environment Concept.
2. Write a simple java program to display **"Welcome all CSE 3rd year Students".**
3. Write a simple program to find out GCD of two numbers.
4. Write a simple java program to display all the prime numbers between 10 and 100.
5. Write a C++ program to enter information of a student using class **Student** and display the information.
6. Write a C++ program to enter the information of the employee using class **Employee** and display the information

DAY 2(CONCEPT OF ARRAY AND CLASS USING CONSTRUCTOR):

1. Write a java program to enter information of a student using class **Student** and display the information. Add constructor and appropriate method to the class.
2. Write a java program to represent a Vector (a series of float values). Include member function to perform following task:
   a. To create the vector
   b. To modify the value of a given element
   c. To multiply the vector by a scalar value
   d. To display the vector

   Add constructor and appropriate method to implement the class.
3. Create a class **Pattern** which will take a variable size array as its data member and display the following pattern. Use constructor and appropriate method to implement the class.

```
1
1 2
1 2 3
1 2 3 4
```

DAY 3(CONCEPT OF CONSTRUTOR OVERLOADING & METHOD OVERLOADING):

1. Create a class called Matrix. It includes methods called 'matMul' for multiplying two Matrix objects and return the resultant Matrix object and 'matAdd' for adding two Matrix objects and return the resultant Matrix
2. Create a class called **Point** which will have 2 co-ordinate value x & y and will be able to calculate the distance between two points. Write appropriate constructor and method of the

class. Create another class called **Shape**, which takes different number of points to create a triangle, rectangle and circle. And also able to calculate the area and perimeter for the different shapes. Write the appropriate overloading method for this shape class. Write appropriate main function to test the shape class.

DAY 4(CONCEPT OF INHERITANCE):

1. Write an abstract class Worker and derive classes DailyWorker and SalariedWorker from it. Every worker has a name and a salary rate. Write an abstract method comPay( ) to compute the weekly pay of every worker. A DailyWorker is paid on the basis of number of days which is to be calculated from the hours he/she works weekly. The SalariedWorker gets paid the wage for a week no matter what actual hours are. Test this program to calculate the weekly pay of workers. Use concept of dynamic polymorphism to write this program.

2. Create a class called point which will have 2 co-ordinate value x & y and will be able to calculate the distance between two points. Write appropriate constructor and method of the class. Create an abstract class called **Shape**, which contains abstract methods to for area(), perimeter(), colour() and compareShape(Shape ob). Create the derived class Triangle, Rectangle and Circle. Write the appropriate overridden methods for these classes. . [compareShape(Shape ob) checks if area and perimeter are same for two objects and return the result[suppose for two Rectangle Objects then return “Matching Rectangles” otherwise return “Non matching Rectangle”]. Test your code using dynamic method dispatch.

DAY 5(CONCEPT OF PACKAGE):

1. Create a package**” BookPack”.** Inside package “BookPack” create a class called **“Book”** which holds the data member publisher, author, publication date. Create another package “BookPackB” and inside this package create another class “**UseBook”** which implements the main function to create an array of object of **“Book”** class and display the details of various book objects.
2. Write a program to make a package **Banking** in which has an abstract class “**Account**” with displayBalance() method in it. Create a class “SavingsAccount extending Account with daily withdrawal limit” of Rs 15000. The minimum balance for “SavingsAccount” is Rs 1000. A penalty of Rs 20 will be deducted on the next deposit if account balance reaches less than the minimum balance. Create another class “LoanAccount” inherited from “SavingsAccount” inside **Loan** package where loan amount will be added to savings account balance and monthly EMI will be deducted from savings account balance. Formula to calculate EMI = P×r×(1 + r)n/((1 + r)n - 1) where, P is Principle Loan Amount, r is rate of interest calculated in monthly basis it should be = Rate of Annual interest/12/100, n is tenure in number of months. [Hint: For 100000 at 10% annual interest for a period of 12 months, r= 10/12/100=0.00833 it comes to 100000*0.00833*(1 + 0.00833)12/((1 + 0.00833)12 - 1) = 8792].

Paper Name:Object Oriented Programming Lab **Paper code: CS594D**

Semester: 5th
Academic Session:

DAY-6(CONCEPT OF INTERFACE & INNER CLASS):

1. Create an interface MessageEncoder that has a single abstract method encode(plainText), where plainText is the message to be encoded. The method will return the encoded message. Create an abstract class Cryptography which takes a plain text (any string) and display the encrypted text .Create two derived classes TranspositionCipher and CaesarCipher that implements the interface MessageEncode. CaesarCipher should have one parameter called shift. Define the method encode() so that each letter is shifted by the value in shift. For example, if shift is 3, a will be replaced by d, b will be replaced by e, c will be replaced by f, and so on. TranspositionCipher have one parameter called key. Define the method encode() so that each letter is transposed by the value of key. For example, if key is 3 1 2, then 1st character will be replaced by 3rd character, 2nd will be replaced by 1st and 3rd will be replaced by 1st. Dispatch the methods dynamically.
2. Write a private inner class **EvenIterato**r that will iterate to the even item of the data structure contains within its enclosing class.

DAY-7(CONCEPT OF EXCEPTION HANDLING):

1. Create an exception class InvalidInputException which takes an integer (say 10) and prints: “InvalidInputException[10]”. Create a class CheckPrime which checks if a given no is prime or not and generate InvaildInputException if inputs are 0, 1 and any negative no. which is derived from a class Number

2. Create a class **Date.** Declare the variables month, day, year as private and create appropriate constructor. Check the condition for exception. If the date is valid, it will proceed the program otherwise display the exception “InvalidDate”. Provide method for checking if the year is leap year or not and method to return the next date. Define method compareTo() to compare two date. Use value returned by compareTo() method to check whether the a date is before or after than other date in methods isBefore() and isAfter()[Use Boolean type for method to return the values as either true or false]. Test your code. [date o/p format: dd-mm-yyyy].

DAY-8(CONCEPT OF STRING CLASS):

1. Write a class StringUtil inside package “strUtil‟ using default constructor and two overloaded constructor one of which receives a String and prints a String that is the exact reversal of the characters in the first String and another constructor takes two arguments-a string and a number- if the number is a valid, then display the character from the string as specified by that number.
2. Using command line argument, Create a class Name_sort which takes names of cities by user and print the name in reverse & sorted order.

Paper Name:Object Oriented Programming Lab **Paper code: CS594D**

Semester: 5th
Academic Session:

DAY-9(CONCEPT OF THREAD):

1. Write a program, which run two threads simultaneously each threads prints "[Hello World]: threadName". Make sure no race condition arises and main thread finishes last.
2. Write a program, which can run two-thread simultaneously. One thread will calculate the factorial of a number and another thread will calculate the factors of that number. Make sure main thread finishes last.
3. Write classes for implementing two threads Writer and Reader, which access a shared integer x. Thread Writer indefinitely write the value of x by increasing it and Reader indefinitely read the value of x by printing its value on the console, i.e threads run in an infinite loop. Make sure Writer must not increase the value of x till Reader is printing and Reader must not print the same value of x. Write the appropriate methods for these two classes.

DAY-10(CONCEPT OF SWING & APPLET):

1. Design a form for displaying your bio data using Java Swing.
2. Create an applet for displaying different geometric shapes using Java Swing.

List of Additional Lab Exercise

Experiment 1

Write a java program to implement a simple socket program using UDP socket.

Experiment 2

Write a java program to implement a bi directional chat program using UDP socket.