Course Name: Object Oriented Programming using Java

Course Code: CSE2121

Program List

Lab 1: Basic Programming

1. Hello World Program
2. Simple Calculator
3. Convert Fahrenheit to Celsius
4. Generate Fibonacci Series
5. Generate Prime Numbers

Lab 2: Flow control and Loops

1. Check if a Number is Even or Odd
2. Use Switch Case to display student grade
3. Calculate Factorial Using a Loop
4. Print Multiplication Tables
5. Calculate the LCM and GCD of Two Numbers

Lab 3: Data types and arrays

1. Write a Java program to calculate the average value of array elements.
2. Write a Java program to find the maximum and minimum value of an array.
3. Write a Java program to addition, subtraction, and multiplication of two matrices of 5x5.
4. Write a Java program to check if the sum of all the 10's in the array is exactly 30. Return false if the condition does not satisfy, otherwise true.
5. Write a Java program to separate even and odd numbers from a given array of integers. Put all even numbers first, and then odd numbers.

Lab 4: Class and object

1. Write a Java program to create a class called "Person" with a name and age attribute. Create two instances of the "Person" class, set their attributes using the constructor, and print their name and age.

2. Create a class named 'Student' with String variable 'name' and integer variable 'roll_no'. Assign the value of roll_no as '2' and that of name as "John" by creating an object of the class Student.

3. Write a Java program to create a class called "Dog" with a name and breed attribute. Create two instances of the "Dog" class, set their attributes using the constructor and modify the attributes using the setter methods and print the updated values.

4. Assign and print the roll number, phone number and address of two students having names "Sam" and "John" respectively by creating two objects of class 'Student'.

5. Write a program to print the area and perimeter of a triangle having sides of 3, 4 and 5 units by creating a class named 'Triangle' without any parameter in its constructor. Also do the same program using parametrized constructor.

Lab 5: Recursion and argument passing

1. Write a Java program to print Fibonacci series of n terms where n is input by user using recursion.
2. Write a Java program to calculate HCF of Two given numbers using recursion.
3. Implement Binary Search Algorithm
4. Write a program to implement object as reference.
5. Write a program to implement object as return type.

Lab 6: Constructor and method overloading

1. Write a program to calculate area of a rectangle using constructor (with no parameter) in java.
2. Write a program to calculate perimeter of a rectangle using constructor (with and without no parameter) in java.
3. Write a program using constructor to display Electricity bill consumer detail by creating a class with the following members: Consumer no., consumer name, previous month reading, current month reading, type of EB connection (i.e domestic or commercial).
4. Write a class Employee that represents an employee of some organization, the class should contain the following members:

• private int id;//Employee id

• private String name;//Employee name

• private int type;//1 = employee, 2 = manager

• private double baseSalary;//Base salary

• public Employee(int _id, String _name);//Constructor

• public void setID(int x);//id mutator

• public void setName(int x);//name mutator

• public intgetID();//id accessor

• public String getName();//name accessor

• public double getSalary(); • //if manager, add 10% to base salary

• public void setBaseSalary(double bs);//sets base salary.

5. Write a class that represents triangle named Triangle, the class must have the following members:

• private double height;//Height

• private double base;//Base length

• public Triangle(double h, double b);//Constructor

• public void setHeight(double x);//Sets height

• public double getHeight();//Gets height

• public void setBase(double x);//Sets base length

• public double getBase();//Gets base length

•public double getArea();//Returns the area of the triangle

Lab 7: Input-output and type casting

1. Reading and Displaying User Input: Create a program that takes user input (e.g., name and age) and displays it, using Scanner class for input and System.out.println for output.

2. Data Conversion: Develop a program that demonstrates explicit type casting, converting a larger data type to a smaller one (e.g., double to int).

3. Arithmetic Calculator: Build a calculator program that takes numeric input from the user, performs arithmetic operations, and displays the result.

4. Temperature Conversion: Create a program that converts temperatures from Celsius to Fahrenheit and vice versa, utilizing user input.

5. BMI Calculator: Develop a program that calculates Body Mass Index (BMI) based on user-entered height and weight values.

Lab 8: Inheritance and overloading

1. Write a Java program to create a class called Vehicle with a method called drive(). Create a subclass called Car that overrides the drive() method to print "Repairing a car".

2. Write a Java program to create a class called Shape with a method called getArea(). Create a subclass called Rectangle that overrides the getArea() method to calculate the area of a rectangle.

3. Write a Java program to create a class called Employee with methods called work() and getSalary(). Create a subclass called HRManager that overrides the work() method and adds a new method called addEmployee().

4. Write a Java program to create a class known as "BankAccount" with methods called deposit() and withdraw(). Create a subclass called SavingsAccount that overrides the withdraw() method to prevent withdrawals if the account balance falls below one hundred.

5. Write a Java program to create a class called Shape with methods called getPerimeter() and getArea(). Create a subclass called Circle that overrides the getPerimeter() and getArea() methods to calculate the area and perimeter of a circle.

Lab 9: Abstract class, Interface, and Packages

1. Create an abstract class Shape with abstract methods for calculating area and perimeter. Implement concrete subclasses (e.g., Circle, Rectangle) that

inherit from Shape and provide their own implementations of these methods.

2. Design an abstract class BankAccount with fields for account number, balance, and abstract methods for deposit and withdrawal. Create concrete subclasses (e.g., SavingsAccount, CheckingAccount) that extend BankAccount and implement the deposit and withdrawal methods.
3. Create an abstract class Employee with attributes like name and salary. Implement subclasses representing different employee types (e.g., Manager, Developer) that inherit from Employee. Define an interface Taxable with a method for calculating taxes, and implement it in employee classes to compute taxes based on salary.
4. Design an interface Sortable with a method to sort an array. Implement this interface in classes representing various sorting algorithms (e.g., BubbleSort, MergeSort).
5. Create an interface Person with methods for getting and setting a person's name. Implement this interface in classes for different roles (e.g., Student, Employee) to manage personal information.

Lab 10: File handling

1. Write a Java program to list all the files in a directory including the files present in all its subdirectories.

2. Write a Java program to read data from the file, we can use subclasses of either InputStream or Reader.

3. WAJP to write sample class Student with following attributes rollno, name and department, create an object and print those details into the file using PrintStream.

4. Write a Java program to use delete method of the file class to delete the specified file or directory ,It return true if the file is deleted and false if the file does not exist.

5. WAJP that reads on file name from the user, then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.

Lab 11: Exception handling

1. Write a Java program that reads a list of integers from the user and throws an exception if any numbers are duplicates.
2. Write a Java program that reads a list of numbers from a file and throws an exception if any of the numbers are positive.
3. Write a Java program to create a method that takes a string as input and throws an exception if the string does not contain vowels.
4. Write a complete java code by writing down the handlers for exceptions thrown by the code. The exceptions the code may throw along with the handler message are listed below:

Division by zero: Print "Invalid division".

String parsed to a numeric variable: Print "Format mismatch".

Accessing an invalid index in string: Print "Index is invalid".

Accessing an invalid index in array: Print "Array index is invalid".

Exceptions other than mentioned above: Any other exception except the above ones fall in this category. Print "Exception encountered".

5. Write a Java program to create a method that reads a file and throws an exception if the file is not found.

Lab 12: Multithread Programming

1. Write a Java program to create and start multiple threads that increment a shared counter variable concurrently.
2. Write a Java program to create a producer-consumer scenario using the wait() and notify() methods for thread synchronization.
3. Write a Java program that creates a bank account with concurrent deposits and withdrawals using threads.
4. Write a Java program to demonstrate Semaphore usage for thread synchronization.
5. Write a Java program that creates a bank account with concurrent deposits and withdrawals using threads.