KIET Group of Institutions, Delhi-NCR, Ghaziabad

Department of Computer Applications

OOP Practical List

S.No	Practical Programs	Scheduled Date	Actual Date	Signature
	UNIT 1: Java Basic Programs:			
1	Write a Java program to insert 3 numbers from keyboard and find out greater number among 3 numbers.			
2	Write a Java program to find out the sum of command line arguments.			
3	Write a Java program to create a Room class, the attributes of this class is roomno, roomtype, roomarea and ACmachine. In this class the member functions are setData and displayData. Use member function to set data and display that data using displayData() method.			
4	Write a Java program to create a class “SimpleObject” and display message by using constructor of this class.			
5	Write a program to demonstrate static variables, methods, and blocks.			
	UNIT 2: Java Object Oriented Programs:			
6	Write a Java program to create a class named Shape and create three sub classes Circle, Triangle and Square, each class has two-member function named draw () and erase (). Implement this concepts using polymorphism.			
7	Write a Java program to give a simple example for abstract class.			
8	Write a Java program to give example for multiple inheritance in Java.			
9	Write a Java program to create class Number with only one private instance variable as a double primitive type, include the following methods isZero(), isPositive(), isNegative( ), isOdd( ), isEven( ), isPrime(), isAmstrong() in this class and all above methods should return boolean primitive type like for isPositive() should return “Positive = True”.			
	UNIT 3: Java Exception Handling:			
10	Write a Java program to illustrate usage of try\catch with finally clause.			
11	Write a Java program to describe usage of throws clause.			
12	Write a Java program for creation of user defined exception.			
13	Create a class Customer having following members: private String custNo private String custName private String category Parameterized constructor to initialize all instance variables Getter methods for all instance variables Perform following validations in the constructor · custNo must start with ‘C’ or ‘c’ · custName must be atleast of 4 characters · category must be either ‘Platinum’, ‘Gold’ or ‘Silver ‘ When any of these validations fail, then raise a user defined exception InvalidInputException Create a class TestCustomer having main method. Ask user to enter customer details. Create an object of Customer and perform validations. Print details of customer.			
14	Write a Java program to create a text file in the path c:\Java\abc.txt and check whether that file exists or not. Using the command exists(), isDirectory(), isFile(), getName() and getAbsolutePath().			
15	Create a class Employee having members as follows: private int empNo private String empName private int empBasic Parameterized constructor to initialize members. Getter methods for all instance variables Create a class WriteEmployee having main method. Ask user to enter details of an employee and set them in an Employee object. Store details of this object in a file emp.txt. Read employee details from the file and display those details.			
	UNIT 4: Java Multi-Threading Programs:			

16	Create a class MyThread derived from Thread class and override the run method. Create a class ThreadDemo having main method. Create 2 objects of MyThread class and observe the behavior of threads			
17	Modify the above to create MyThread class by implementing Runnable interface and observe the behavior of threads.			
18	Assign different priorities to the 2 threads and observe the behaviour			
19	Implement three classes: Storage, Counter and Printer The Storage class should store an integer. The Counter class should create a thread and starts counting from 0 (0,1,2, 3 ...) and stores each value in the Storage class. The Printer class should create a thread that keeps reading the value in the Storage class and printing it. Write a program that creates an instance of the Storage class and set up a Counter and Printer object to operate on it.			
	UNIT 5: Java GUI Applications:			
20	Design and implement a GUI for the Temperature class. One challenge of this design is to find a good way for the user to indicate whether a Fahrenheit or Celsius value is being input. This should also determine the order of the conversion: F to C or C to F. [C/5 = (F -32)/9]			
21	Design and implement a GUI application for Calculator same as Windows OS calculator.			
22	Design and create small notepad type application using Swing/AWT.			