CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

Faculty of Technology and Engineering

U & P U. Patel Department of Computer Engineering

Date : 27 / 06 / 2022

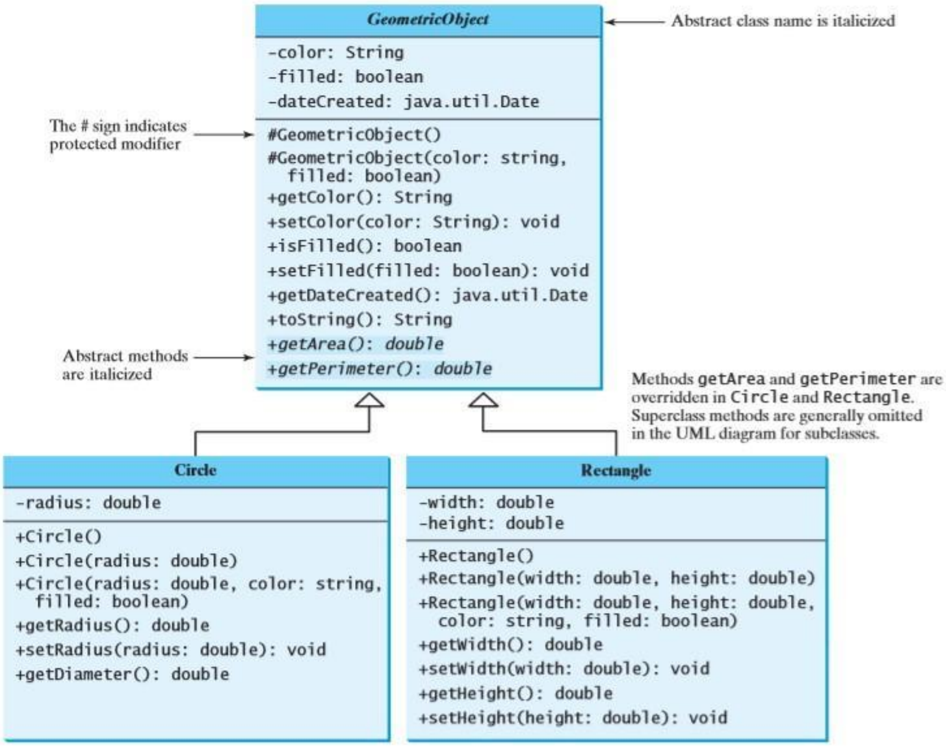
Practical List

Academic Year	:	2022-23	Semester	:	3
Course code	:	CE251	Course name	:	Java Programming

Sr No	Aim	CO
	PART-I **Data Types, Variables, Arrays, Operators, Control Statements, String**	
1.	Introduction to Object Oriented Concepts, comparison of Java with other object oriented programming languages. Introduction to JDK, JRE, JVM, javadoc, command line argument.	1
2.	A typical mobile number in India is “+91-AA-BBB-CCCCC”. Where the first two digits (AA) indicate a mobile system operator, the next three (BBB) denote the mobile switching code (MSC) while the remaining five digits (CCCCC) are unique to the subscriber. Write an application that takes a mobile number as an input from a user in above mentioned format and display code for mobile system operator, mobile switching code and last 5 digits which are unique to subscriber. Ex. For an input +91-94-999-65789, output should be : Mobile system operator code is 94 MSC is 999 Unique code is 65789	1
3.	Given two non-negative int values, return true if they have the same first digit, such as with 72 and 75. firstDigit(7, 71) → true firstDigit(6, 17) → false firstDigit(31, 311) → true	1

4.	The problem is to write a program that will grade multiple-choice tests. Assume there are eight students and ten questions, and the answers are stored in a two-dimensional array. Each row records a student's answers to the questions, as shown in the following array. Students' Answers to the Questions: 0 1 2 3 4 5 6 7 8 9 Student 0 A B A C C D E E A D Student 1 D B A B C A E E A D Student 2 E D D A C B E E A D Student 3 C B A E D C E E A D Student 4 A B D C C D E E A D Student 5 B B E C C D E E A D Student 6 B B A C C D E E A D Student 7 E B E C C DE E A D The key is stored in a one-dimensional array: Key to the Questions: 0 1 2 3 4 5 6 7 8 9 Key D B D C C D A E A D Your program grades the test and displays the result. It compares each student's answers with the key, counts the number of correct answers, and displays it.	1
5.	We have triangle made of blocks. The topmost row has 1 block, the next row down has 2 blocks, the next row has 3 blocks, and so on. Compute recursively (no loops or multiplication) the total number of blocks in such a triangle with the given number of rows. triangle(0) → 0 triangle(1) → 1 triangle(2) → 3	1
	PART-II **Object Oriented Programming: Classes, Methods, Inheritance**	**CO**
1.	Design a class named Cylinder containing following attributes and behavior. ● One double data field named radius. The default value is 1. ● One double data field named height. The default value is 1. ● A no-argument constructor that creates a default Cylinder. ● A Single argument constructor that creates a Cylinder with the specified radius. ● Two argument constructor that creates a Cylinder with the specified radius and height. ● A method named getArea() that returns area of the Cylinder. ● Create a class TestCylinder and test and display result.	1,2
2.	Design a class named Account that contains: ● A private int data field named id for the account (default 0). ● A private double data field named balance for the account (default 500₹). ● A private double data field named annualInterestRate that stores the current interest rate (default 7%). Assume all accounts have the same interest rate. ● A private Date data field named dateCreated that stores the date when the account was created. ● A no-arg constructor that creates a default account. ● A constructor that creates an account with the specified id and initial balance. ● The accessor and mutator methods for id, balance, and annualInterestRate. ● The accessor method for dateCreated. ● A method named getMonthlyInterestRate() that returns the monthly interest rate. ● A method named getMonthlyInterest() that returns the monthly interest. ● A method named withdraw that withdraws a specified amount from the account.	1,2

	● A method named deposit that deposits a specified amount to the account.	
3.	Use the Account class created as above to simulate an ATM machine. Create 10 accounts with id AC001….AC010 with initial balance 300₹. The system prompts the users to enter an id. If the id is entered incorrectly, ask the user to enter a correct id. Once an id is accepted, display menu with multiple choices. **1.** Balance inquiry **2.** Withdraw money [Maintain minimum balance 300₹] **3.** Deposit money **4.** Money Transfer **5.** Create Account **6.** Deactivate Account **7.** Exit Hint: Use ArrayList, which is can shrink and expand with compared to Array.	1,2
4.	(Subclasses of Account) In Programming Exercise 2, the Account class was defined to model a bank account. An account has the properties account number, balance, annual interest rate, and date created, and methods to deposit and withdraw funds. Create two subclasses for checking and saving accounts. A checking account has an overdraft limit, but a savings account cannot be overdrawn. Write a test program that creates objects of Account, SavingsAccount, and CheckingAccount and invokes their toString() methods.	1,2
5.	Develop a Program that illustrate method overloading concept.	1,2
PART-III **Package & Interface**		**CO**
1.	Create an abstract class GeometricObject as the superclass for Circle and Rectangle. GeometricObject models common features of geometric objects. Both Circle and Rectangle contain the getArea() and getPerimeter() methods for computing the area and perimeter of a circle and a rectangle. Since you can compute areas and perimeters for all geometric objects, so define the getArea() and getPerimeter() methods in the GeometricObject class. Give implementation in the specific type of geometric object. Create TestGeometricObject class to display area and perimeter of Rectangle and Triangle, compare area of both and display results. Design of all classes are given in the following UML diagram.	1,2

GeometricObject
-color: String
-filled: boolean
-dateCreated: java.util.Date
#GeometricObject()
#GeometricObject(color: string, filled: boolean)
+getColor(): String
+setColor(color: String): void
+isFilled(): boolean
+setFilled(filled: boolean): void
+getDateCreated(): java.util.Date
+toString(): String
+getArea(): double
+getPerimeter(): double
Abstract class name is italicized
The # sign indicates protected modifier
Abstract methods are italicized
Methods getArea and getPerimeter are overridden in Circle and Rectangle. Superclass methods are generally omitted in the UML diagram for subclasses.
Circle
-radius: double
+Circle()
+Circle(radius: double)
+Circle(radius: double, color: string, filled: boolean)
+getRadius(): double
+setRadius(radius: double): void
+getDiameter(): double
Rectangle
-width: double
-height: double
+Rectangle()
+Rectangle(width: double, height: double)
+Rectangle(width: double, height: double, color: string, filled: boolean)
+getWidth(): double
+setWidth(width: double): void
+getHeight(): double
+setHeight(height: double): void

2.	Write a program to create a default method in an interface IPrinter. Create an interface IPrinter and IScanner. You can assume variables and methods for both interfaces. Create a concrete class to implement both the interfaces. Create 5 objects of the class, store it in Vector and display the result of the vector.	1,2
3.	WAP that illustrate the interface inheritance. Interface P is extended by P1 and P2 interfaces. Interface P12 extends both P1 and P2. Each interface declares one method and one constant. Create one class that implements P12. By using the object of the class invokes each of its method and displays constant.	1,2
4.	Develop a Program that illustrate method overriding concept.	1,2
5.	Write a java program which shows importing of classes from other user define packages.	1,2
6.	Write a program that demonstrates use of packages & import statements.	1,2
7.	Write a program that illustrates the significance of interface default method.	1,2
	PART-IV **Exception Handling**	
1.	WAP to show the try - catch block to catch the different types of exception.	4
2.	WAP to generate user defined exception using “throw” and “throws” keyword.	4
3.	Write a program that raises two exceptions. Specify two ‘catch’ clauses for the two exceptions. Each ‘catch’ block handles a different type of exception. For example the exception could be ‘ArithmeticException’ and ‘ArrayIndexOutOfBoundsException’. Display a message in the ‘finally’ block.	4
	PART-V File Handling & Streams	**CO**
1.	WAP to show how to create a file with different mode and methods of File class to find path, directory etc.	4

2.	When to use Character Stream over Byte Stream? When to use Byte Stream over Character Stream? Give example.	4
3.	Write a program to transfer data from one file to another file so that if the destination file does not exist, it is created.	4
4.	WAP to show use of character and byte stream.	4
5.	Write a program to enter any 15 numbers from the user and store only even numbers in a file named “Even.txt”. And display the contents of this file on the console. (BufferedReader / BufferedWriter).	4
6.	WAP to demonstrate methods of wrapper class.	4
PART-VI Multithreading		**CO**
1.	Write a program to create thread which display “Hello World” message. A. by extending Thread class B. by using Runnable interface.	3
2.	Generate 15 random numbers from 1 to 100 and store it in an int array. Write a program to display the numbers stored at odd indexes by thread1 and display numbers stored at even indexes by thread2.	3
3.	Write a program to increment the value of one variable by one and display it after one second using thread using sleep() method.	3
4.	Write a program to create three threads ‘FIRST’, ‘SECOND’, ‘THIRD’. Set the priority of the ‘FIRST’ thread to 3, the ‘SECOND’ thread to 5(default) and the ‘THIRD’ thread to 7.	3
5.	Write a program to solve producer-consumer problem using thread Synchronization.	3
	PART-VII **Collection Framework and Generic**	**CO**
1.	Create a generic method for sorting an array of Comparable objects.	5,6
2.	Write a program that counts the occurrences of words in a text and displays the words and their occurrences in alphabetical order of the words. Using Map and Set Classes.	5.6
3.	Personal Loan Eligibility Criteria for Salaried Applicant is as follows: Eligible Age Group - 21 years to 60 years Minimum Net Monthly Income - Rs. 15,000 Minimum Total Work Experience - 1 year Citizenship – Indian Create a class AccountHolder to store above given information entered by a user. Create 5 objects of AccountHolder class and store them in an ArrayList. Display names of account holders , who are eligible to get a loan based on given criteria.	5,6