|  |  |  |
| --- | --- | --- |
| **Kongu-Black** | **Kongu Engineering College, Perundurai- 638 060**  **Department of Information Technology** | TRANSFORM YOURSELF |

**Unit II**

|  |  |
| --- | --- |
| Course Code : 22ITC31 | Year / Semester :II/III |
| Course Name : Java Programming | Faculty Name : Dr.S.Subashini |

**Tutorial 8**

**Question Bank**

**Part B**

|  |  |  |
| --- | --- | --- |
|  | Develop a Java Program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.  abstract class Shape {  protected int dimension1;  protected int dimension2;  public Shape(int dimension1, int dimension2) {  this.dimension1 = dimension1;  this.dimension2 = dimension2;  }  public abstract void printArea();  }  class Rectangle extends Shape {  public Rectangle(int length, int width) {  super(length, width);  }  public void printArea() {  int area = dimension1 \* dimension2;  System.out.println("Area of Rectangle: " + area);  }  }  class Triangle extends Shape {  public Triangle(int base, int height) {  super(base, height);  }  public void printArea() {  double area = 0.5 \* dimension1 \* dimension2;  System.out.println("Area of Triangle: " + area);  }  }  class Circle extends Shape {  public Circle(int radius) {  super(radius, 0);  }  public void printArea() {  double area = Math.PI \* dimension1 \* dimension1;  System.out.println("Area of Circle: " + area);  }  }  public class Main {  public static void main(String[] args) {  Rectangle rectangle = new Rectangle(5, 10);  Triangle triangle = new Triangle(3, 8);  Circle circle = new Circle(6);  rectangle.printArea();  triangle.printArea();  circle.printArea();  }  } | **(CO2, K3)** |
|  | Construct an abstract class ‘Employeeabstract’ with attributes name and employeeId and method calculateSalary(). Create an Employee that implements the abstract class methods and additionally display() method.  abstract class EmployeeAbstract {  protected String name;  protected int employeeId;  public EmployeeAbstract(String name, int employeeId) {  this.name = name;  this.employeeId = employeeId;  }  public abstract void calculateSalary();  // Additional method in the abstract class  public void display() {  System.out.println("Employee ID: " + employeeId);  System.out.println("Employee Name: " + name);  }  }  class Employee extends EmployeeAbstract {  private double salary;  public Employee(String name, int employeeId, double salary) {  super(name, employeeId);  this.salary = salary;  }  // Implementation of the abstract method  public void calculateSalary() {  System.out.println("Calculating salary for Employee...");  // Perform salary calculation logic (this is a placeholder)  System.out.println("Salary: " + salary);  }  }  public class Main {  public static void main(String[] args) {  // Creating an instance of the Employee class  Employee employee = new Employee("John Doe", 1001, 50000.0);  // Calling methods from the abstract class and the implemented class  employee.display();  employee.calculateSalary();  }  } | **(CO2, K3)** |
|  | Construct an abstract class ‘Studentabstract’ with attributes name and rollno and method calculateGrade(). Create a Student class that implements the abstract class methods and additionally display() method.  abstract class StudentAbstract {  protected String name;  protected int rollNo;  public StudentAbstract(String name, int rollNo) {  this.name = name;  this.rollNo = rollNo;  }  public abstract void calculateGrade();  public void display() {  System.out.println("Roll Number: " + rollNo);  System.out.println("Student Name: " + name);  }  }  class Student extends StudentAbstract {  private char grade;  public Student(String name, int rollNo, char grade) {  super(name, rollNo);  this.grade = grade;  }  // Implementation of the abstract method  public void calculateGrade() {  System.out.println("Calculating grade for Student...");  // Perform grade calculation logic (this is a placeholder)  System.out.println("Grade: " + grade);  }  }  public class Main {  public static void main(String[] args) {  // Creating an instance of the Student class  Student student = new Student("Alice Smith", 101, 'A');  // Calling methods from the abstract class and the implemented class  student.display();  student.calculateGrade();  }  } | **(CO2, K3)** |
|  | Develop an Interest interface which contains simpleInterest() and compInterest() methods and static final field of Rate 25%. Write a class to implement those methods.  // Interest interface  interface Interest {  // Abstract methods  double simpleInterest(double principal, double time);  double compInterest(double principal, double rate, double time);  // Static final field  double RATE = 0.25; // 25% interest rate  }  // Class implementing the Interest interface  class InterestCalculator implements Interest {  public double simpleInterest(double principal, double time) {  // Simple Interest formula: SI = P \* R \* T / 100  return (principal \* RATE \* time) / 100.0;  }  public double compInterest(double principal, double rate, double time) {  // Compound Interest formula: CI = P \* (1 + R/n)^(nt) - P  int n = 12; // Assuming interest is compounded monthly  double compoundInterest = principal \* Math.pow(1 + rate / (n \* 100), n \* time) - principal;  return compoundInterest;  }  }  public class Main {  public static void main(String[] args) {  InterestCalculator calculator = new InterestCalculator();  // Calculate simple interest  double principal = 1000.0;  double time = 2.5;  double simpleInterest = calculator.simpleInterest(principal, time);  System.out.println("Simple Interest: $" + simpleInterest);  // Calculate compound interest  double rate = RATE; // Using the static final field from the interface  double compoundInterest = calculator.compInterest(principal, rate, time);  System.out.println("Compound Interest: $" + compoundInterest);  }  } | **(CO2, K3)** |
|  | Develop an Employee interface which contains claculatesalary() and display() methods. Create a manager class that implements these methods.  // Employee interface  interface Employee {  // Abstract methods  void calculateSalary();  void display();  }  // Manager class implementing the Employee interface  class Manager implements Employee {  private String name;  private int employeeId;  private double salary;  public Manager(String name, int employeeId, double salary) {  this.name = name;  this.employeeId = employeeId;  this.salary = salary;  }  public void calculateSalary() {  // Manager salary calculation logic (this is a placeholder)  System.out.println("Calculating salary for Manager...");  System.out.println("Salary: $" + salary);  }  public void display() {  System.out.println("Employee ID: " + employeeId);  System.out.println("Employee Name: " + name);  }  }  public class Main {  public static void main(String[] args) {  // Creating an instance of the Manager class  Manager manager = new Manager("John Doe", 1001, 80000.0);  // Calling methods from the Employee interface  manager.display();  manager.calculateSalary();  }  } | **(CO2, K3)** |
|  | Create a Java interface named Student with methods get() and display(). Create a info class and marks class to implement these methods.  **import** java.util.Scanner;  **interface** student {  **void** get();  **void** display();  }  **class** info **implements** student {  String name, rollno;  **public** **void** get() {  Scanner sc=**new** Scanner(System.*in*);  System.*out*.println("student name and roll number");  name=sc.next();  rollno=sc.next();  }  **public** **void** display() {  System.*out*.println("Name: " + name);  System.*out*.println("Rollno: " + rollno);  }    **void** show() {  System.*out*.println("student basic detail");  }  }  **class** marks **implements** student {  **int** m1,m2,m3,total,average;  **public** **void** get() {  Scanner sc=**new** Scanner(System.*in*);  System.*out*.println("enter three subject marks");  m1=sc.nextInt();  m2=sc.nextInt();  m3=sc.nextInt();  }  **void** calc() {  total=m1+m2+m3;  average=total/3;  }  **public** **void** display() {  System.*out*.println("Mark1: " + m1 + " Mark2: " + m2 + " Mark3: " + m3);  System.*out*.println("Total: " + total);  System.*out*.println("Average: " + average);  }  }  **class** sample {  **public** **static** **void** main(String[] args) {  student s;  info i = **new** info();  marks m = **new** marks();  s=i;  s.get();  i.show();  s.display();  s=m;  s.get();  m.calc();  s.display();  }  } | **(CO2, K3)** |