Java Programming [CSE201] Enrolment No.:23DCS100

CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY

**DEVANG PATEL INSTITUTE OF ADVANCE TECHNOLOGY & RESEARCH**

Department of Computer Science & Engineering

Subject Name: Java Programming

Semester: III

Subject Code: CSE201

Academic year: 2024-25

Part - 4

|  |  |
| --- | --- |
| **No.** | **Aim of the Practical** |
| 17. | Create a class with a method that prints "This is parent class" and its subclass with another method that prints "This is child class". Now, create an object for each of the  class and call 1 - method of parent class by object of parent.  **PROGRAM CODE:**  *class* parent{  *public* void display(){  System.out.println("This is parent class");  }  }  *class* child *extends* parent{  *public* void cdisplay(){  System.out.println("This is child class");  }  }    *public* *class* JAVA\_P17 {  *public* *static* void main(String[] args) {  parent p1=new parent();  p1.display();  child c1=new child();  c1.cdisplay();  }  }  **OUTPUT:**    **CONCLUSION:**  This code demonstrates basic inheritance in Java. The `parent` class has a method `display()`, while the `child` class, which extends `parent`, adds its own method `cdisplay()`. In the `JAVA\_P17` class, an object of `parent` calls the `display()` method, and an object of `child` calls the `cdisplay()` method. This shows how inheritance allows the child class to extend the functionality of the parent class. |
| 18. | Create a class named 'Member' having the following  members: Data members  1 - Name  2 - Age  3 - Phone number  4 - Address  5 – Salary  It also has a method named 'printSalary' which prints the  salary of the members. Two classes 'Employee' and  'Manager' inherits the 'Member' class. The 'Employee' and  'Manager' classes have data members 'specialization' and  'department' respectively. Now, assign name, age, phone  number, address and salary to an employee and a manager  by making an object of both of these classes and print the  same.  **PROGRAM CODE:**  import *java.util.Scanner*;  *class* Member {  String name;  int age;  long phone\_number;  String address;  int salary;  void getData(Scanner sc) {  System.out.print("Enter the name: ");  name = sc.next();  System.out.print("Enter the age: ");  age = sc.nextInt();  System.out.print("Enter the phone number: ");  phone\_number = sc.nextLong();  sc.nextLine(); *// Consume the newline character left by nextLong()*  System.out.print("Enter the address: ");  address = sc.nextLine();  System.out.print("Enter the salary: ");  salary = sc.nextInt();  }  void putData() {  System.out.println("Employee's name         : " + name);  System.out.println("Employee's age          : " + age);  System.out.println("Employee's Phone number : " + phone\_number);  System.out.println("Employee's address      : " + address);  }  void printSalary() {  System.out.println("Employee's salary       : " + salary);  }  }  *class* Employee *extends* Member {  String specialization;  @*Override*  void getData(Scanner sc) {  super.getData(sc);  System.out.print("Enter the specialization: ");  specialization = sc.next();  }  @*Override*  void putData() {  super.putData();  System.out.println("Specialization is       : " + specialization);  }  }  *class* Manager *extends* Member {  String department;  @*Override*  void getData(Scanner sc) {  super.getData(sc);  System.out.print("Enter the department: ");  department = sc.next();  }  @*Override*  void putData() {  super.putData();  System.out.println("Department is           : " + department);  }  }  *public* *class* JAVA\_P18 {  *public* *static* void main(String[] args) {  Scanner sc = new Scanner(System.in);  System.out.println("1) Employee \n2) Manager");  int n = sc.nextInt();  switch (n) {  case 1:  Employee e1 = new Employee();  e1.getData(sc);  System.out.println();  e1.putData();  e1.printSalary();  break;  case 2:  Manager m1 = new Manager();  m1.getData(sc);  System.out.println();  m1.putData();  m1.printSalary();  break;  default:  System.out.println("Invalid choice.");  break;  }  sc.close();  }  }  **OUTPUT:**    **CONCLUSION:**  This Java program uses inheritance to manage employee and manager data. The `Member` class provides basic details and salary methods, while `Employee` and `Manager` classes extend it with additional specifics. The `JAVA\_P18` class allows the user to input and display data for either an employee or a manager based on their choice. |
| 19. | Create a class named 'Rectangle' with two data members 'length' and 'breadth' and two methods to print the area and perimeter of the rectangle respectively. Its constructor  having parameters for length and breadth is used to initialize length and breadth of the rectangle. Let class 'Square' inherit the 'Rectangle' class with its constructor having a parameter for its side (suppose s) calling the constructor of its parent class as 'super(s,s)'. Print the area and perimeter of a rectangle and a square. Also use array of objects.  **PROGRAM CODE:**  *class* Rectangle {  float length;  float breadth;  *public* Rectangle(float length, float breadth) {  this.length = length;  this.breadth = breadth;  }  void areaOfRectangle() {  System.out.println("Area of rectangle is : " + (this.length \* this.breadth));  }  void perimeterOfRectangle() {  System.out.println("Perimeter of rectangle is : " + (2 \* (this.length + this.breadth)));  }  }  *class* Square *extends* Rectangle {  *public* Square(float side) {  super(side, side);  }  @*Override*  void areaOfRectangle() {  System.out.println("Area of square is : " + (this.length \* this.length));  }  @*Override*  void perimeterOfRectangle() {  System.out.println("Perimeter of square is : " + (4 \* this.length));  }  }  *public* *class* JAVA\_P19 {  *public* *static* void main(String[] args) {  Rectangle[] shapes = new Rectangle[2];  shapes[0] = new Rectangle(12, 24);  shapes[1] = new Square(10);  for (Rectangle shape : shapes) {  shape.areaOfRectangle();  shape.perimeterOfRectangle();  System.out.println();  }  }  }  **OUTPUT:**    **CONCLUSION:**  This Java program demonstrates polymorphism with a `Rectangle` base class and a `Square` subclass. It creates an array of `Rectangle` objects, initializes it with both `Rectangle` and `Square` instances, and uses overridden methods to calculate and display the area and perimeter for each shape. |
| 20. | Create a class named 'Shape' with a method to print "This is This is shape". Then create two other classes named 'Rectangle', 'Circle' inheriting the Shape class, both having a method to print "This is rectangular shape" and "This is circular shape" respectively. Create a subclass  'Square' of 'Rectangle' having a method to print "Square is a rectangle". Now call the method of 'Shape' and 'Rectangle' class by the object of 'Square' class.  **PROGRAMCODE:**  *class* shape {  void print(){  System.out.println("This is This is shape ");  }  }  *class* Rectangle *extends* shape{  void print1(){  System.out.println("This is rectangular shape");  }  }  *class* Circle *extends* shape{  void print2(){  System.out.println("This is circular shape");  }  }  *class* Square *extends* Rectangle{  void print3(){  System.out.println("Square is a rectangle");  }  }    *public* *class* JAVA\_20 {  *public* *static* void main(String[] args) {  Square s1=new Square();  s1.print();  s1.print1();  }  }  **OUTPUT:**    **CONCLUSION :**  This Java program defines a class hierarchy with `shape` as the base class and `Rectangle`, `Circle`, and `Square` as subclasses. In the `main` method, a `Square` object is created, and it demonstrates inheritance by calling methods from both `shape` and `Rectangle` classes. |
| 21. | Create a class 'Degree' having a method 'getDegree' that prints "I got a degree". It has two subclasses namely 'Undergraduate' and 'Postgraduate' each having a method with the same name that prints "I am an Undergraduate" and "I am a Postgraduate" respectively. Call the method by creating an object of each of the three classes.  **PROGRAM CODE :**  *class* Degree{  void getdegree(){  System.out.println("I got a degree ");  }  }  *class* Undergraduate *extends* Degree{  void getDegree(){  System.out.println("I am an Undergraduate");  }  }  *class* Postgraduate *extends* Degree{  void getDegree(){  System.out.println("I am a Postgraduate");  }  }  *public* *class* JAVA\_P21 {  *public* *static* void main(String[] args) {  Degree d1 = new Degree();  Undergraduate u1 = new Undergraduate();  Postgraduate p1 = new Postgraduate();  d1.getdegree();  u1.getDegree();  p1.getDegree();  }  }  **OUTPUT:**    **CONCLUDION:**  This Java program defines a `Degree` class with a method `getDegree`, and subclasses `Undergraduate` and `Postgraduate` each with their own `getDegree` methods. The `main` method creates objects of each class and calls their respective methods to demonstrate method overriding. |
| 22. | Write a java that implements an interface AdvancedArithmetic which contains amethod signature int divisor\_sum(int n). You need to write a class called MyCalculator which implements the interface. divisorSum function just takes an integer as input and  return the sum of all its divisors. For example, divisors of 6 are 1, 2, 3 and 6, so  divisor\_sum should return 12. The value of n will be at most 1000.  **PROGRAM CODE :**  import *java.util.Scanner*;  *interface* *AdvancedArithmetic* {  int divisor\_sum(int n);  }  *class* JAVA\_P22 *implements* *AdvancedArithmetic* {  *public* int divisor\_sum(int n) {  int sum = 0;  for (int i = 1; i <= n; i++) {  if (n % i == 0) {  sum += i;  }  }  return sum;  }  *public* *static* void main(String[] args) {  JAVA\_P22 myCalculator = new JAVA\_P22();  int number ;  System.out.print("Enter the number : ");  Scanner sc= new Scanner(System.in);  number=sc.nextInt();  int result = myCalculator.divisor\_sum(number);  System.out.println("Sum of divisors of " + number + " is: " + result);  }  }  **OUTPUT:**    **CONCLUSION:**  This Java program implements the `AdvancedArithmetic` interface with a `divisor\_sum` method that calculates the sum of all divisors of a given number. The `JAVA\_P22` class provides the implementation and computes the result based on user input. The `main` method reads an integer from the user, computes the sum of its divisors, and prints the result. |
| 23. | Assume you want to capture shapes, which can be either circles (with a radiusand a color) or rectangles (with a length, width, and color). You also want to be able to create signs (to post in the campus center, for example), each of which has a shape (for the background of the sign) and the text (a String) to put on the sign. Create classes and interfaces for circles, rectangles, shapes, and signs. Write a program that illustrates the significance of interface default method.  **PROGARM CODE:**  *interface* *Shape* {  double area();  String color();  String shape();  *default* void setDescription() {  System.out.println("This is a "+shape()+ " with " + color() + " color and area " + area());  }  }  *class* Circle *implements* *Shape* {  double radius;  String color;  *public* Circle(double radius, String color) {  this.radius = radius;  this.color = color;  }  *public* double area() {  return 3.14 \* radius \* radius;  }  *public* String color() {  return color;  }  *public* String shape(){  return "Circle";  }  }  *class* Rectangle *implements* *Shape* {  double length;  double width;  String color;  *public* Rectangle(double length, double width, String color) {  this.length = length;  this.width = width;  this.color = color;  }  *public* double area() {  return length \* width;  }  *public* String color() {  return color;  }  *public* String shape(){  return "Rectangle";  }  }  *class* Sign {  *Shape* shape;  String text;  *public* Sign(*Shape* shape, String text) {  this.shape = shape;  this.text = text;  }  *public* void display() {  System.out.println("Sign text: " + text);  shape.setDescription();  }  }  *public* *class* JAVA\_P23 {  *public* *static* void main(String[] args) {  *Shape* circle = new Circle(5.0, "Red");  *Shape* rectangle = new Rectangle(4.0, 6.0, "Blue");  Sign circleSign = new Sign(circle, "Welcome to the Circle Zone");  Sign rectangleSign = new Sign(rectangle, "Welcome to the Rectangle Zone");  circleSign.display();  rectangleSign.display();  }  }  **OUTPUT:**    **CONCLUSION:**  This Java program uses the `Shape` interface with methods for calculating area and providing color and shape details. The `Circle` and `Rectangle` classes implement this interface, each providing specific implementations. The `Sign` class associates a `Shape` with a text message and displays both. The `main` method creates instances of `Circle` and `Rectangle`, creates signs for each, and prints their descriptions along with their areas and colors. |