**CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY**

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

Department of Computer Science & Engineering

**Subject Name: Java Programming**

**Semester: 3rd**

**Subject Code: CSE201**

**Academic year: 2024-2025**

**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 :**  import java.util.\*;  class parent  {  void display()  {  System.out.println("This is parent class.");  }  }  class child extends parent  {  void display1()  {  System.out.println("This is child class.");  }  }  public class prc17  {  public static void main(String args[])  {  parent p=new parent();  child c=new child();  p.display();  }  }  **OUTPUT:**    **CONCLUSION:**  By this code I learnt the basic concept of inheritance and how to implement it. |
| **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.\*;  class member  {  String name;  int age;  String num;  String address;  long salary;    void printSalary()  {  System.out.println("The salary is: "+salary);  }  }  class Employee extends member  {  String specialization;  Employee()  {}  Employee(String n, int a, String no, String add, long s, String spec)  {  name=n;  age=a;  num=no;  address=add;  salary=s;  specialization=spec;  //printSalary();  }  void print()  {  System.out.println("Name: "+name);  System.out.println("Age: "+age);  System.out.println("Number: "+num);  System.out.println("Address: "+address);  System.out.println("Specialization: "+specialization);  printSalary();  }  }  class Manager extends member  {  String dept;  Manager()  {}  Manager(String n, int a, String no, String add, long s, String d)  {  name=n;  age=a;  num=no;  address=add;  salary=s;  dept=d;  //printSalary();  }  void print()  {  System.out.println("Name: "+name);  System.out.println("Age: "+age);  System.out.println("Number: "+num);  System.out.println("Address: "+address);  System.out.println("Department: "+dept);  printSalary();  }  }  public class prc18  {  public static void main(String args[])  {  Employee e1=new Employee("JUGVI",20,"7575068948","Laxminarayan vihar society",50000,"Cloud computing");  System.out.println("The employee details are: ");  e1.print();    System.out.println();    Manager m1=new Manager("SHAILI DESAI",40,"8369463746","20, Aristro villa",75000,"Machine learning");  System.out.println("The manager details are: ");  m1.print();  }  }  **OUTPUT:**    **CONCLUSION:**  By this program, we learnt about inheritance and overriding methods in child classes. |
| **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:**  import java.util.\*;  class rectangle  {  double length,breadth;    rectangle()  {}    rectangle(double l, double b)  {  length=l;  breadth=b;  }    void area()  {  System.out.println("The area is: "+length\*breadth);  }    void perimeter()  {  System.out.println("The perimeter is: "+2\*(length+breadth));  }  }  class square extends rectangle  {  square()  {}  square(double s)  {  super(s,s);  }  void area()  {  System.out.println("The area is: "+length\*breadth);  }    void perimeter()  {  System.out.println("The perimeter is: "+4\*length);  }  }  public class prc19  {  public static void main(String args[])  {  rectangle[] shapes=new rectangle[3];  shapes[0] = new rectangle(2, 6);  shapes[1] = new square(2);  shapes[2] = new rectangle(4, 8);    for(rectangle shape : shapes)  {  if(shape instanceof square)  {  System.out.println("For square: ");  }  else  {  System.out.println("For rectangle: ");  }  shape.area();  shape.perimeter();  System.out.println();  }  }  }  **OUTPUT:**    **CONCLUSION:**  By this experiment I learnt how to implement single inheritance. |
| **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.  **PROGRAM CODE:**  import java.util.\*;  class shape  {  void print()  {  System.out.println("This is a 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 prc20  {  public static void main(String args[])  {  square s=new square();  s.print();  s.print1();  }  }  **OUTPUT:**    **CONCLUSION:**  Hereby, we learn about multi-level inheritance and implementing the concepts of classes and objects. By creating an object of the Square class and calling methods from both Shape and Rectangle, the code illustrates how subclasses inherit and can extend the functionality of their parent 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:**  import java.util.\*;  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 prc21  {  public static void main(String args[])  {  Degree d=new Degree();  Undergraduate u=new Undergraduate();  Postgraduate p=new Postgraduate();  d.getDegree();  u.getDegree();  p.getDegree();  }  }  **OUTPUT:**    **CONCLUSION:**  By this code, we learn about hierarchical inheritance and making the objects of each class to implement the given declared methods respectively. |
| **22.** | Write a java that implements an interface AdvancedArithmetic which contains amethod signature int divisor\_sum(int n). You need to write a class calledMyCalculator 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.\*;  interface AdvancedArithmetic  {  int divisorSum(int n);  }  class MyCalculator implements AdvancedArithmetic  {  public int divisorSum(int n)  {  int sum=0;  for(int i=1;i<=n;i++)  {  if(n%i==0)  {  sum+=i;  }  }  return sum;  }  }  public class prc22  {  public static void main(String args[])  {  MyCalculator m=new MyCalculator();  Scanner s=new Scanner(System.in);  System.out.print("Enter the number you want to find the sum of divisors for: ");  int n=s.nextInt();  System.out.println("The sum of its divisors is: "+m.divisorSum(n));  }  }  **OUTPUT:**    **CONCLUSION:**  By this program, we learn about declaring interfaces and implementing them in another class which gives us the total sum of all the divisors of a particular number which is taken from the user. |
| **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.  **PROGRAM CODE:**  **OUTPUT:**  **CONCLUSION:** |