CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY

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

Department of Computer Science & Engineering

Subject Name:

Semester:

Subject Code:

Academic year:

Part - 3

|  |  |
| --- | --- |
| **No.** | **Aim of the Practical** |
| 12. | Imagine you are developing a currency conversion tool for a  travel agency. This tool should be able to convert an amount  in Pounds to Rupees. For simplicity, we assume the  conversion rate is fixed: 1 Pound = 100 Rupees. The tool  should be able to take input both from command-line  arguments and interactively from the user.  **PROGRAM CODE:**  import java.util.Scanner;  public class pra12a {      public static void main(String []args){          Scanner sc = new Scanner(System.in);          System.out.println("enter value in pound");          float pound = sc.nextFloat();          float ruppe = pound \* 100;          System.out.println("amount in ruppe" + ruppe);            sc.close();      }  }  **OUTPUT:**    **CONCLUSION:**  This program effectively converts a given value in pounds to rupees by multiplying the input value by 100. It utilizes the Scanner class to take user input and performs the conversion calculation. The result is then displayed to the user, providing a straightforward and practical example of basic arithmetic operations and user interaction in Java. This simple yet functional program demonstrates the use of standard input and output operations, making it a useful tool for beginners learning Java. |
| 13. | Create a class called Employee that includes three pieces of information as instance variables—a first name (type String), a last name (type String) and a monthly salary  (double). Your class should have a constructor that initializes the three instance variables. Provide a set and a get method for each instance variable. If the monthly salary  is not positive, set it to 0.0. Write a test application named EmployeeTest that demonstrates class Employee’s capabilities. Create two Employee objects and display each  object’s yearly salary. Then give each Employee a 10% raise and display each Employee’s yearly salary again.  **PROGRAM CODE:**  import java.util.\*;  public class pra13 {      public static void main(String[] args) {          Scanner sc = new Scanner(System.in);          System.out.println("enter the FName, LName and Salary of employee 1");          String FName = sc.next();          String LName = sc.next();          int msalary = sc.nextInt();          Employee emp1 = new Employee(FName, LName, msalary);          System.out.println(" initial yearly salary of " + emp1.FName + " " + emp1.LName + " is " + emp1.getYearlySalary());          System.out.println(" updated yearly salary of " + emp1.FName + " " + emp1.LName + " is " + emp1.getYearlySalary()\*1.10);          sc.close();      }  }  **OUTPUT:**    **CONCLUSION:**  This program captures and processes employee details, including first name, last name, and monthly salary. It calculates and displays the initial yearly salary and the updated yearly salary with a 10% increase. By utilizing the Scanner class for input and performing basic arithmetic operations, the program demonstrates practical applications of Java for handling employee salary data. This example is useful for understanding object-oriented programming and basic input/output operations in Java.  11of30 |

|  |  |
| --- | --- |
| 14. | Create a class called Date that includes three pieces of information as instance variables—a month (type int), a day (type int) and a year (type int). Your class should have a constructor that initializes the three instance variables and assumes that the values provided are correct. Provide a set and a get method for each instance variable. Provide a  method displayDate that displays the month, day and year separated by forward slashes (/). Write a test application named DateTest that demonstrates class Date’s capabilities.  **PROGRAM CODE:**  import java.util.Scanner;  class Date {      private      int date;      int month;      int year;      Date(int date,int month,int year){          this.date=date;          this.month=month;          this.year=year;      }      public int getdate(){          return date;      }      public void setdate(int date){          this.date=date;      }      public int getmonth(){          return month;      }      public void setmonth(int month){          this.month=month;      }      public int getyear(){          return year;      }      public void setyear(int year){          this.year=year;      }      public void Display(){          System.out.println(date + "/" + month + "/" + year);      }  }  class practical14{      public static void main(String[] args) {          Scanner sc = new Scanner(System.in);          System.out.print("Enter date :");          int date= sc.nextInt();          System.out.print("Enter month :");          int month= sc.nextInt();          System.out.print("Enter year :");          int year= sc.nextInt();          Date d = new Date(date , month , year);          d.Display();          sc.close();      }  }  **OUTPUT:**    **CONCLUSION:**  This program captures and displays a date in the format dd/mm/yyyy. It uses a Date class with private data members for the day, month, and year, along with getter and setter methods to access and modify these values. The Display method prints the date in the desired format. The practical14 class demonstrates the functionality by taking user input for the date, month, and year, creating a Date object, and displaying the date. This example illustrates the principles of encapsulation and object-oriented programming in Java.  12of30 |
| 15. | Write a program to print the area of a rectangle by creating  a class named 'Area' taking the values of its length and  breadth as parameters of its constructor and having a  method named 'returnArea' which returns the area of the  rectangle. Length and breadth of rectangle are entered  through keyboard.  **PROGRAM CODE:**  import java.util.Scanner;  class Area {      Area(int length,int breadth)      {          int a=length;          int b=breadth;      }       static int returnArea(int length,int breadth)      {          int area=length\*breadth;          return area;      }  }  public class Practical15  {  public static void main(String args[])  {       Area ar;       Scanner sc= new Scanner(System.in);      System.out.print("Enter the value of length: ");      int a= sc.nextInt();      System.out.print("Enter the value of breadth: ");      int b=sc.nextInt();      System.out.println("Area of Rectangle: " + Area.returnArea(a,b) );      sc.close();  }  }  **OUTPUT:**    **CONCLUSION:**  This program calculates the area of a rectangle by taking the length and breadth as input from the user. The Area class contains a static method returnArea that computes the area. The Practical15 class demonstrates the functionality by prompting the user for the dimensions, calling the returnArea method, and displaying the result. This example highlights the use of static methods and basic arithmetic operations in Java, providing a clear and practical application of object-oriented programming concepts.  13of30 |

|  |  |
| --- | --- |
| 16. | Print the sum, difference and product of two complex  numbers by creating a class named ‘Complex’ with separate  methods for each operation whose real and imaginary parts  are entered by user.  **PROGRAM CODE:**  import java.util.Scanner;  class Complex {      void sum(int r1,int i1,int r2,int i2)      {          int r=r1+r2;          int i=i1+i2;          System.out.println("Sum of two numbers are: "+ r +" + " + i +" i ");      }      void difference(int r1,int i1,int r2,int i2)      {          int r=r1-r2;          int i=i1-i2;          System.out.println("Subtraction of two numbers are: "+ r +" + " + i +" i ");      }      void multiplication(int r1,int i1,int r2,int i2)      {          int r= (r1\*r2-i1\*i2);          int i=(r1\*i2+i1\*r2);          System.out.println("Multiplication  of two numbers are: "+ r +" + " + i +" i ");      }  }  public class Practical16  {      public static void main(String args[])      {           Complex c = new Complex();           Scanner sc=new Scanner(System.in);          System.out.println("Enter the first number ");          System.out.print("Enter the real part of the complex number: ");          int r1=sc.nextInt();          System.out.print("Enter the imaginary part of the complex number: ");          int i1=sc.nextInt();          System.out.println("Enter the second number ");          System.out.print("Enter the real part of the complex number: ");          int r2=sc.nextInt();          System.out.print("Enter the imaginary part of the complex number: ");          int i2=sc.nextInt();          c.sum(r1,i1,r2,i2);          c.difference(r1,i1,r2,i2);          c.multiplication(r1,i1,r2,i2);          sc.close();      }  }  **OUTPUT:**    **CONCLUSION:**  This program performs arithmetic operations on complex numbers, including addition, subtraction, and multiplication. The Complex class contains methods for each operation, and the Practical16 class demonstrates the functionality by taking user input for two complex numbers and displaying the results of the operations. This example highlights the use of object-oriented programming to handle complex number arithmetic, showcasing the practical application of methods and user interaction in Java. |