**PROGRAM NO : 10**

**AIM:** Area of different shapes using overloaded functions.

**ALGORITHM :**

Step 1: Start

Step 2: Define the main class

Step 3: Define methods with the same methodname that performs the area operation for each shape

Step 4: Display the areas of each shapes.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Areas.java | package javaprg;  import java.util.Scanner;  public class shapes {  public void area(int l,int b) {  System.out.println("Area of reactangle is:"+l\*b);  }    public void area(float r)  {  System.out.println("Area of circle is:"+3.14\*r\*r);  }    public static void main(String[] args) {  int l,b;  float r;  Scanner obk=new Scanner(System.in);  shapes obj=new shapes();  System.out.println("Enter the length and breadth of reactangle:");  l=obk.nextInt();  b=obk.nextInt();  System.out.println("Enter the radius of the circle:");  r=obk.nextInt();  obj.area(l,b);  obj.area(r);    }  } |

**OUTPUT:**

Graphical user interface, text, application

Description automatically generated

**PROGRAM NO : 11**

**AIM:** Create a class ‘Employee’ with data members Empid, Name, Salary, Address and constructors to initialize the data members. Create another class ‘Teacher’ that inherit the properties of class employee and contain its own data members department, Subjects taught and constructors to initialize these data members and also include display function to display all the data members. Use array of objects to display details of N teachers.

**ALGORITHM :**

Step 1: Start

Step 2: create class “employee” with the provided data members and define the constructors

Step 3: create another class “Teachers” that performs inheritance of employee class and define constructors for the same

Step 4: create an array of objects in the corresponding class

Step 5: Display the details for the number of teachers provided.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Employee.java | import java.util.Scanner;  class employees{  int b;  String c;  String a;  int s;  public employees() {  Scanner obq=new Scanner(System.in);  Scanner obq1=new Scanner(System.in);Scanner obq2=new Scanner(System.in);  System.out.println("Enter the employee N.O :");  b=obq.nextInt();  System.out.println("Enter the employee Name :");  c=obq1.nextLine();  System.out.println("Enter the employee Address :");  a=obq2.nextLine();  System.out.println("Enter the employee salary:");  s=obq.nextInt();  }  public void edisplay() {  System.out.println("Employee N.O :"+b);  System.out.println("Employee Name :"+c);  System.out.println("Employee Address :"+a);  System.out.println("Employee salary:"+s);  }  }  class teachers extends employees {  String c,a;  public teachers() {  Scanner obq1=new Scanner(System.in);  Scanner obq2=new Scanner(System.in);  System.out.println("Enter the employee deparment :");  c=obq1.nextLine();  System.out.println("Enter the employee subject :");  a=obq2.nextLine();  }  void tdisplay() {  System.out.println("Employee deparment :"+c);  System.out.println("Employee subject :"+a);  }  }  class data extends teachers{  public data() {  }  public void display(int i) {  System.out.println("\n"+"----------"+i+"st Employee ---------"+"\n");  tdisplay();  edisplay();  }  }  public class employee{  public static void main(String[] args) {  int a;  Scanner obq=new Scanner(System.in);  System.out.println("Enter the number of employee :");  a=obq.nextInt();  data[] obj=new data[a];  for(int i=0;i<a;i++)  {  obj[i]=new data();  }  for(int i=0;i<a;i++)  {  obj[i].display(i+1);  }  }  } |

**OUTPUT:**

Text

Description automatically generated

Text

Description automatically generated

**PROGRAM NO : 12**

**AIM:** Create a class ‘Person’ with data members Name, Gender, Address, Age and a constructor to initialize the data members and another class ‘Employee’ that inherits the properties of class Person and also contains its own data members like Empid, Company\_name, Qualification, Salary and its own constructor. Create another class ‘Teacher’ that inherits the properties of class Employee and contains its own data members like Subject, Department, Teacherid and also contain constructors and methods to display the data members. Use array of objects to display details of N teachers.

**ALGORITHM :**

STEP 1: Start

STEP 2: Create a class person with data members name, gender, address, age  
STEP 3: Create constructor to initialize these object  
STEP 4: Create another class Employee that inherited the properties of class person and also creates its data members like empid, companyname , qualification, salary and its constructor  
STEP 5: Create another class Teacher that inherits the properties of class employee and creates its data members  
STEP 6: Creates array of object  
STEP 7: Using these arrays of objects print the values  
STEP 8: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Employee.java | import java.util.\*;  public class person {  int Age;  String Address,Name,Gender;  public person()  {  Scanner n = new Scanner(System.in);  System.out.println("Enter Name : ");  Name = n.next();  System.out.println("Enter Age : ");  Age = n.nextInt();  System.out.println("Enter Gender : ");  Gender = n.next();  System.out.println("Enter Address : ");  Address = n.next();  }  }  class emp extends person  {  int eid,salary;  String Qual,comp\_name;  public emp()  {  Scanner m = new Scanner(System.in);  System.out.println("Enter Emp\_id : ");  eid = m.nextInt();  System.out.println("Enter Salary : ");  salary = m.nextInt();  System.out.println("Enter Qualification : ");  Qual = m.next();  System.out.println("Enter Company Name : ");  comp\_name = m.next();  }  }  class teacher extends emp  {  int tid;  String dept,subj;  teacher()  {  Scanner o= new Scanner(System.in);  System.out.println("Enter Teacher\_id : ");  tid = o.nextInt();  System.out.println("Enter Subject: ");  subj = o.next();  System.out.println("Enter Department : ");  dept = o.next();  }  public void disp()  {  System.out.println("Name : "+Name);  System.out.println("Age : "+Age);  System.out.println("Gender : "+Gender);  System.out.println("Address : "+Address);  System.out.println("Emp id : "+eid);  System.out.println("salary : "+salary);  System.out.println("Qualification : "+Qual);  System.out.println("Company Name : "+comp\_name);  System.out.println("Teacher id : "+tid);  System.out.println("Subject : "+subj);  System.out.println("Department : "+dept);  }  public static void main(String args[])  {  int no;  Scanner p = new Scanner(System.in);  System.out.println("Enter no of teachers : ");  no = p.nextInt();  teacher[] obj = new teacher[no];  for(int i=0;i<no;i++)  {  obj[i] = new teacher();  }  for(int i=0;i<no;i++)  {  System.out.println("\nDetails of Employee No : " + (i+1));  obj[i].disp();  }  }  } |

**OUTPUT:**

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

**PROGRAM NO : 13**

**AIM:** Write a program has class Publisher, Book, Literature and Fiction. Read the information and print the details of books from either the category, using inheritance

**ALGORITHM :**

STEP 1: Start

STEP 2: Create a class publisher, Book, Literature and Fiction using inheritance  
STEP 3: Create each class’s data members and member functions  
STEP 4: Read this information from the user  
STEP 5: Print the details of book from the category  
STEP 6: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Book.java | package myproj;  import java.util.\*;  class publisher  {  String pbname;  Scanner a=new Scanner(System.in);  public publisher()  {    System.out.println("Enter publisher name:");  pbname=a.next();  }  }  class book extends publisher  {  String bookname;  String Author;  public book()  {  System.out.println("Enter book name:");  bookname=a.next();    System.out.println("Enter Author name:");  Author=a.next();      }  }  class literature extends book  {  public literature()  {  System.out.println("category:literature");  System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");  }  void display()  {  System.out.println("publishername:"+pbname);  System.out.println("book name:"+bookname);  System.out.println("Author name:"+Author);  }  }  class fiction extends book  {  public fiction()  {  System.out.println("category:fiction");  System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");  System.out.println("\n");  }  void display()  {  System.out.println("publishername:"+pbname);  System.out.println("book name:"+bookname);  System.out.println("Author name:"+Author);  }  }  public class books {    public static void main(String[] args) {    int i,nb;    Scanner b=new Scanner(System.in);  System.out.println("Enter the no of literature books you need to store:");  nb=b.nextInt();  literature l[]=new literature[nb];  for( i=0;i<nb;i++)  {  l[i]=new literature();    }  System.out.println("Enter the no of fictional books you need to store:");  int m;  m=b.nextInt();  fiction f[]=new fiction[m];  for( i=0;i<m;i++)  {  f[i]=new fiction();    }    System.out.println("\_\_\_Displaying literature books:\_\_");  for(i=0;i<nb;i++)  {  System.out.println("Displaying details of book no"+(i+1));  l[i].display();  }  System.out.println("\_\_\_Displaying fictional books:\_\_");  for(i=0;i<m;i++)  {  System.out.println("Displaying details of book no:"+(i+1));  f[i].display();  }      }  } |

**OUTPUT:**

Text

Description automatically generated

Text

Description automatically generated

**PROGRAM NO : 14**

**AIM:** Create classes Student and Sports. Create another class Result inherited from Student and Sports. Display the academic and sports score of a student.

**ALGORITHM :**

STEP 1: Start

STEP 2: Create classes student and sports  
STEP 3: Create another class Result inherited from Student and Sports  
STEP 4: Create a function called display to print  
STEP 5: display academic and sports score of student  
STEP 6: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| sport.java | package myprj;  import java.util.\*;  class Student{  int rollno;  String Name;  float sub1,sub2,sub3;  Student(){  Scanner sc=new Scanner(System.in);  System.out.println("Please input the student details. .\n");  System.out.println("Enter the Roll No:");  rollno = sc.nextInt();  System.out.println("Enter the Name:");  Name = sc.next();  System.out.println("\nInput the Subject marks. . \n");  System.out.println("Enter marks for Subject-1:");  sub1 = sc.nextFloat();  System.out.println("Enter marks for Subject-2:");  sub2 = sc.nextFloat();  System.out.println("Enter marks for Subject-3:");  sub3 = sc.nextFloat();  }  }  class Sports extends Student{  String event1,event2;  int score1,score2;  Sports(){  Scanner sc=new Scanner(System.in);  System.out.println("\nInput the Sports data. . \n");  System.out.println("Enter Event-1:");  event1 = sc.next();  System.out.println("Enter score:");  score1 = sc.nextInt();  System.out.println("Enter Event-2");  event2 = sc.next();  System.out.println("Enter score:");  score2 = sc.nextInt();    sc.close();  }  }  class Result extends Sports{  void Display() {  System.out.println("\_\_\_STUDENT PROFILE\_\_\_");  System.out.println("Roll No: "+rollno);  System.out.println("Name: "+Name);    System.out.println("\n\_\_\_ACADEMIC SCORE\_\_\_");  System.out.println("Subject-1 Marks: "+sub1);  System.out.println("Subject-2 Marks: "+sub2);  System.out.println("Subject-3 Marks: "+sub3);  System.out.println("TOTAL:"+(sub1+sub2+sub3));    System.out.println("\n\_\_\_SPORTS SCORE\_\_\_");  System.out.println("Event-1: "+event1);  System.out.println("Score: "+score1);  System.out.println("Event-2: "+event2);  System.out.println("Score: "+score2);  System.out.println("TOTAL:"+(score1+score2));    }  }  public class Score{  public static void main(String[] args) {    Result r = new Result();  System.out.println("\n\_\_\_\_\_\_\_\_\_\_REPORT CARD\_\_\_\_\_\_\_\_\n");  r.Display();    }  } |

**OUTPUT:**

Text

Description automatically generated

Text

Description automatically generated

**PROGRAM NO : 15**

**AIM:** Create an interface having prototypes of functions area() and perimeter(). Create two classes Circle and Rectangle which implements the above interface. Create a menu driven program to find area and perimeter of objects.

**ALGORITHM :**

Step 1 : Start

Step 2: create interfaces with area and perimeter as prototypes

Step 3: create class circle

Step 4: get the radius

Step 5: implement the interface

Step 7: create class rectangle

Step 8: get the length and breadth

Step 9: implement the interface

Step 10: display the area and perimeter

Step 11: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Area.java | package myprj;  import java.util.Scanner;  import java.lang.\*;  interface calculation{  void data\_input();  void area();  void perimeter();  }  class Circle implements calculation{  float r;    @Override  public void data\_input() {    Scanner sc = new Scanner(System.in);  System.out.println("\nEnter the radius of Circle:");  r = sc.nextFloat();    }    @Override  public void area() {  System.out.println("->Area of Circle:"+(Math.PI\*r\*r));  }    @Override  public void perimeter() {  System.out.println("->Perimeter of Circle:"+(2\*Math.PI\*r));  }  }  class Rectangle extends Circle{  int l,b;    @Override  public void data\_input() {  super.data\_input();  Scanner sc1 = new Scanner(System.in);  System.out.println("Enter the length of Rectangle:");  l = sc1.nextInt();  System.out.println("Enter the breadth of Rectangle:");  b = sc1.nextInt();    }    @Override  public void area() {  super.area();  System.out.println("->Area of Rectangle:"+(l\*b));  }    @Override  public void perimeter() {  super.perimeter();  System.out.println("->Perimeter of Rectangle:"+(2\*(l+b)));  }    }  public class Interface\_menu {  public static void main(String[] args) {  Rectangle r = new Rectangle();    int choice;  boolean input = false;  Scanner sc2 = new Scanner(System.in);  System.out.println("\_\_\_\_AREA & PERIMETER CALCULATOR\_\_\_");    while(input == false){  System.out.println("\n\_\_\_\_M A I N M E N U\_\_\_\n1.To input the values \t2.To find area \n3.To find perimeter\t4.Exit\nEnter your Choice:");  choice = sc2.nextInt();    switch(choice) {    case 1:  r.data\_input();  break;  case 2:  r.area();  break;  case 3:  r.perimeter();  break;  case 4:  System.out.println("EXITING . . .");  System.exit(0);    }    }  // TODO Auto-generated method stub  }  } |

**OUTPUT:**

Text

Description automatically generated

Text

Description automatically generated

**PROGRAM NO : 16**

**AIM:** Prepare bill with the given format using calculate method from interface :

Order No.

Date :

Product Id Name Quantity unit price Total

-----------------------------------------

101 A 2 25 50

102 B 1 100 100

-----------------------------------------

Net. Amount 150

**ALGORITHM :**

Step 1: Start

Step 2: get the number of items

Step 3: get the item id,unit price ,quantity

Step 4: calculate the net amount using interface

Step 5: display the bill

Step 6: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Bill.java | package opsproj;  import java.util.\*;  import java.time.\*;  interface calBill  {  public void calc();    }  class calculate implements calBill  {  int id,qnt,up,tot,on,date;  String name;  public void inform()  {  Scanner inf = new Scanner(System.in);    System.out.println("Enter Product ID : ");  id = inf.nextInt();  System.out.println("Enter Name : ");  name = inf.next();  System.out.println("Enter Quantity : ");  qnt = inf.nextInt();  System.out.println("Unit Price : ");  up = inf.nextInt();    }    public void orderdetails()  {  Scanner ord = new Scanner(System.in);  System.out.println("Enter Order No : ");  on = ord.nextInt();  }    public void calc()  {  tot = qnt \* up;  System.out.println(" "+id+"\t\t "+name+"\t "+qnt+"\t\t "+up+"\t "+tot);      }    void display()  {  System.out.println("Order No : " + on);  LocalDate obj = LocalDate.now();  System.out.print("\n");  System.out.println("Date : " + obj);  System.out.print("\n");  System.out.println("Product Id\tName\tQuantity\tunit price\tTotal");  System.out.print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");  System.out.print("\n");    }  }  public class Q7BILL  {  public static void main(String[] args)  {  int no,net = 0;    Scanner inp3 = new Scanner(System.in);  System.out.println("Enter no of products : ");  no = inp3.nextInt();  calculate or = new calculate();    calculate obj[] = new calculate[no];    or.orderdetails();    for(int i=0;i<no;i++)  {  System.out.print("\n");  System.out.println("Enter Details of product "+ (i+1));  System.out.println("==========================");  obj[i] = new calculate();  obj[i].inform();    }      or.display();    for(int i=0;i<no;i++)  {  obj[i].calc();  net = net + obj[i].tot;  }    System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");  System.out.println(" "+"\t\t "+"\t "+"\t\tNet amount :"+"\t "+net);      }  } |

**OUTPUT:**

Table

Description automatically generated

**PROGRAM NO : 17**

**AIM:** Create a Graphics package that has classes and interfaces for figures Rectangle, Triangle, Square and Circle. Test the package by finding the area of these figures.

**ALGORITHM :**

STEP 1 : Start

STEP 2: Define a package contain functions to assign calculate and display .

STEP 3: Read inputs from user and assign values to objects.

STEP 4: Perform desired operations.

STEP 5: Print the Outputs.

STEP 6: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| 1.java | package Graphics;  import java.util.Scanner;  interface calculation{  public void rectangle();  public void triangle();  public void circle();  public void square();  }  public class Dimension implements calculation {  double area;  Scanner obj1=new Scanner(System.in);  public void rectangle() {  int l,b;  System.out.println("Enter the length of the rectangle:");  l=obj1.nextInt();  System.out.println("Enter the Breath of the rectangle:");  b=obj1.nextInt();  area=l\*b;  System.out.println("Area:"+area+"\n");    }  public void triangle() {  int h,b;  System.out.println("Enter the base of the Triangle:");  b=obj1.nextInt();  System.out.println("Enter the height of the Triangle:");  h=obj1.nextInt();  area=(h\*b)/2;  System.out.println("Area:"+area+"\n");  }  public void circle() {  float r;  System.out.println("Enter the Radius of the Circle:");  r=obj1.nextInt();    area=3.14\*r\*r;  System.out.println("Area:"+area+"\n");  }  public void square() {  int s;  System.out.println("Enter the Side of the Square:");  s=obj1.nextInt();    area=s\*s;  System.out.println("Area:"+area+"\n");  }  } |
| Main.java | package Main;  import java.util.Scanner;  import Graphics.Dimension;  public class Test {  public static void main(String[] args) {  Scanner obk=new Scanner(System.in);  Dimension obj=new Dimension();  int ch=0;      while(ch<5) {  System.out.println("Choose the Shape to find the area:"+"\n 1.Rectangle \t 2.Triangle \n 3.Circle \t 4.Sqaure \t 5.Exit");  ch=obk.nextInt();  switch(ch) {    case 1:  obj.rectangle();  break;  case 2:  obj.triangle();  break;  case 3:  obj.circle();  break;  case 4:  obj.square();  break;  case 5: break;  default:  System.out.println("invalid choice");  break;    }  }  }} |

**OUTPUT:**

Text

Description automatically generated

**PROGRAM NO : 18**

**AIM:** Create an Arithmetic package that has classes and interfaces for the 4 basic arithmetic operations. Test the package by implementing all operations on two given numbers.

**ALGORITHM :**

STEP 1 : Start

STEP 2: Define a package contain functions to assign calculate and display .

STEP 3: Read inputs from user and assign values to objects.

STEP 4: Perform desired operations.

STEP 5: Print the Outputs.

STEP 6: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| cO4q2.java | package c04q2;  //Create an Arithmetic package that has classes and interfaces for the 4 basic arithmetic  //operations. Test the package by implementing all operations on two given numbers  import arithmetic.operations;  import java.util.Scanner;  public class C04Q2  {  public static void main(String[] args)  {  double n1,n2;  int ch;  operations ob = new operations();  Scanner sc = new Scanner(System.in);  System.out.println("Enter the 2 numbers:");  n1 = sc.nextDouble();  n2 = sc.nextDouble();  System.out.println("\n1.Addition\n2.Subtraction\n3.Multiplication\n4.Division\nEnter the choice:");  ch = sc.nextInt();  switch(ch)  {  case 1:ob.addition(n1,n2);  break;  case 2:ob.subtraction(n1,n2);  break;  case 3:ob.multiplication(n1,n2);  break;  case 4:ob.division(n1,n2);  break;  default:System.out.println("Invalid choice");  }  }    } |
| Arithmetic.java | package arithmetic;  interface basic  {  void addition(double a,double b);  void subtraction(double a,double b);  void multiplication(double a,double b);  void division(double a,double b);  }  public class operations implements basic  {  @Override  public void addition(double a, double b) {  System.out.println(a+" + "+b+" = "+(a+b));  }  @Override  public void subtraction(double a, double b) {  System.out.println(a+" - "+b+" = "+(a-b));  }  @Override  public void multiplication(double a, double b) {  System.out.println(a+" x "+b+" = "+(a\*b));  }  @Override  public void division(double a, double b) {  System.out.println(a+" / "+b+" = "+(a/b));  }    } |

**OUTPUT:**

Graphical user interface, text

Description automatically generated

**PROGRAM NO : 19**

**AIM:** Write a user defined exception class to authenticate the user name and password.

**ALGORITHM :**

STEP 1 : Start

STEP 2: Read inputs as username and password.

STEP 3: Verify the username and password.

STEP 4: If its true; Print Succesful message:.

STEP 5: Else print error meessage.

STEP 6: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Javaprograms.java | package javaprograms;  //Write a user defined exception class to authenticate the user name and password  import java.util.Scanner;  class authException extends Exception  {  public authException(String s) {  super(s);  }    }  public class CO4Q3  {  public static void main(String[] args) {  String username = "admin";  String passcode = "pass1234";  String user\_name,password;  Scanner sc = new Scanner(System.in);  try  {  System.out.println("Enter the username:");  user\_name = sc.nextLine();  // sc.nextLine();  System.out.println("Enter the password:");  password = sc.nextLine();  if(username.equals(user\_name) && passcode.equals(password))  {  System.out.println("Authentication successful...");  }  else  throw new authException("Invalid user credentials");    }  catch(authException e)  {  System.out.println("Exception caught "+e);  }  }  } |

**OUTPUT:**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**PROGRAM NO : 20**

**AIM:** Find the average of N positive integers, raising a user defined exception for each negative input.

**ALGORITHM :**

STEP 1 : Start

STEP 2: Enter a limit n.

STEP 3:Read n elements.

STEP 4:iterate the loop .

STEP 5:if i>0;. then

STEP 6: sum=sum+i

STEP 7: Else Print error message

STEP 8: Av=sum/n

STEP 9: Print av

STEP 10: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Javaprograms.java | package javaprograms;  //Find the average of N positive integers, raising a user defined exception for each negative  //input  import java.util.Scanner;  class NegativeIntegerException extends Exception  {  public NegativeIntegerException(String s)  {  super(s);  }  }  public class CO4Q4 {  public static void sample()  {  try {  int n,count=0;  float num[];  float total=0;  Scanner sc = new Scanner(System.in);  System.out.print("Enter the number of values:");  n = sc.nextInt();  num = new float[n];  System.out.println("Enter the numbers:");  for(int i=0;i<n;i++)  {  num[i] = sc.nextInt();  try{  if(num[i]<0)  {  throw new NegativeIntegerException("Negative integer");  }  else  {  total += num[i];  count++;  }  }catch(NegativeIntegerException e)  {  System.out.println("Exception caught "+e);  }  }  System.out.println("Average = "+(total/count));  } catch (Exception e) {  System.out.println("Exception caught "+e);  }  }  public static void main(String[] args) {  try {  sample();  } catch (Exception e) {  }  }  } |

**OUTPUT:**

Graphical user interface, text, application, email

Description automatically generated

**PROGRAM NO : 21**

**AIM:** Define 2 classes; one for generating multiplication table of 5 and other for displaying first N prime numbers. Implement using threads. (Thread class)

**ALGORITHM :**

STEP 1 : Start

STEP 2: Define a class for multiplication and another for prime numbers.

STEP 3:Read a input as n.

STEP 4:Print the multiplication table of n.

STEP 5: Print n prime numbers.

STEP 6: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| PrimeNumbers.java | import java.util.Scanner;  import java.lang.\*;  import java.io.\*;  public class Q5 {  public static void main(String[] args) throws InterruptedException {  multiplication\_tbl a = new multiplication\_tbl();  prime\_num b = new prime\_num();  a.start();  a.sleep(200);  b.start();  b.sleep(200);    }  }  class multiplication\_tbl extends Thread{  public void run(){  int n=10;  Scanner sc= new Scanner(System.in);  System.out.println("multiplication table of 5");  for(int i=1;i<=10;i++)  {  System.out.println(+i+"x 5 ="+(i\*5));  }  }  }  class prime\_num extends Thread{  public void run() {  int n,i=3,count,m;  Scanner sc= new Scanner(System.in);  System.out.println("Enter limit for prime numbers:");  n=sc.nextInt();    if(n >=1)  {  System.out.println("First prime" +n+ "numbers are:");  System.out.println("2");  for(count = 2; count <= n; i++)  {    for(m= 2; m < i;m++)  {  if(i%m == 0)  break;  }  if(m== i)  {  System.out.println(i);  count++;  }  }  }  }  } |

**OUTPUT:**

Text

Description automatically generated

**PROGRAM NO : 22**

**AIM:** Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface).

**ALGORITHM :**

STEP 1 : Start

STEP 2: Define a class for Fibonacci and another for even numbers.

STEP 3:Read a input as n.

STEP 4:Print the Fibonnaci series of n number.

STEP 5: Printn n even numbers.

STEP 6: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| EvenSeries.java | import java.util.Scanner;  import java.lang.\*;  import java.io.\*;  public class Q6 {  public static void main(String[] args) throws InterruptedException {  fibonacci at = new fibonacci();  Thread a=new Thread(at);  even bt = new even();  Thread b= new Thread(bt);  a.start();  a.sleep(200);  b.start();  b.sleep(200);  }  }  class fibonacci extends Thread{  public void run(){  int n=20, a1=0, a2=1;  Scanner sc= new Scanner(System.in);  System.out.println("Fibonacci Series till " + n + " terms:");  for (int i=1;i<=20;i++) {  System.out.print(a1 + ", ");  int nextTerm = a1+a2;  a1=a2;  a2=nextTerm;  }  }  }  class even extends Thread{  public void run() {  int number, i;  Scanner sc = new Scanner(System.in);  System.out.print("\n\n Please Enter any Number : ");  number = sc.nextInt();  for(i = 1; i <= number; i++)  {  if(i % 2 == 0)  {  System.out.print(i +"\n");  }  }    }  } |

**OUTPUT:**

Graphical user interface, text, application

Description automatically generated

**PROGRAM NO : 23**

**AIM:** Producer/Consumer using ITC

**ALGORITHM :**

STEP 1 : Start

STEP 2: Define a class name as product with members pname,pcode and price.

STEP 3: Define objects to Class and add 3 products and values to each data using the object.

STEP 4: Check whether which product has lowest price using if-else ststement.

STEP 5: Print the details of product.

STEP 6: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.ArrayList;  import java.util.List;  class Producer extends Thread{  List<Integer> list;    public Producer(List<Integer> list){  this.list=list;  }    @Override  public void run(){  try {    synchronized(list){    while(true){    if(list.size()>0) {    list.wait();  }  else  produce();  }  }  }  catch(InterruptedException e) {  e.printStackTrace();  }  }  private void produce() throws InterruptedException {  for(int i=1;i<=5;i++){  Thread.sleep(1000);  list.add(i);  System.out.println("Producer produced the element: "+i);  }  list.notifyAll();  }  }  class Consumer extends Thread{  List<Integer> list;    public Consumer(List<Integer> list){  this.list=list;  }    @Override  public void run(){  try {    while(true){    synchronized(list){    if(list.size()==0){    list.wait();  }  else  consume();  }  }  }  catch(InterruptedException e)  {  e.printStackTrace();  }  }  private void consume() throws InterruptedException  {  while(!list.isEmpty())  {  Thread.sleep(1000);  System.out.println("Consumed the element: "+list.remove(0));  }  list.notifyAll();  }  }  public class co4\_q7  {  public static void main(String[] args) throws Exception  {  List<Integer> list = new ArrayList<Integer>();  Producer produce = new Producer(list);  Consumer consume = new Consumer(list);  produce.start();  consume.start();  }  } |

**OUTPUT :**

Table

Description automatically generated

**PROGRAM NO : 24**

**AIM:** Program to create a generic stack and do the Push and Pop operations.

**ALGORITHM :**

STEP 1 : Start

STEP 2: Define a stack .

STEP 3: Push(); enter an element n

If top<n

Top++

Stack[top]=n.

STEP 4:Pop()

If top!=1

.top--

STEP 5: Print the stack.

STEP 6: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.Scanner;  public class Q8 {  int top=-1,ch,item,i;  int a[] = new int[10];  Scanner sc = new Scanner(System.in);  public static void main(String[] args) {  Q8 obj = new Q8 ();  obj.stack();  }  public void stack(){  while(ch<3) {  System.out.println("Enter the size of the array:");  int N=sc.nextInt();  System.out.println("\t CHOICES : ");  System.out.println("\n 1.push \n 2.pop \n 3.exit \n");  System.out.println("\n Enter your choice:");  ch=sc.nextInt();  switch(ch){  case 1:  System.out.println("Enter the element to be inserted:");  item=sc.nextInt();  if(top==N-1) {  System.out.println("stack overflow!");  }  else {  top++;  a[top]=item;  }  break;  case 2:  if(top==-1) {  System.out.println("stack is empty");  }  else {  item=a[top];  top--;  System.out.println("deleted element is:" +item);  }  break;  case 3 : break;  default : System.out.println("\n Invalid choice");  }  if(top < 0){  System.out.println("\n stack is empty");  }  else{  System.out.println("\n stack is \n");  for(i=top;i>=0;i--){  System.out.println(a[i]);  }  }  }  }  } |

**OUTPUT :**

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**PROGRAM NO : 25**

**AIM:** Using generic method perform Bubble sort.

**ALGORITHM :**

STEP 1 : Start

STEP 2: Look at the first number in the list.

STEP 3: Compare the current number with the next number.

STEP 4: Is the next number smaller than the current number? If so, swap the two numbers around. If not, do not swap.

STEP 5: Move to the next number along in the list and make this the current number.

STEP 6: Repeat from step 2 until the last number in the list has been reached.

STEP 7: If any numbers were swapped, repeat again from step 1.

STEP 8: If the end of the list is reached without any swaps being made, then the list is ordered and the algorithm can stop.

STEP 9: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | package myproj;  import java.util.\*;  public class Bubblsrt {  public static void main(String[] args) {  int num, i, j, temp;  Scanner x = new Scanner(System.in);    System.out.println("Enter the number of integers to sort:");  num = x.nextInt();    int array[] = new int[num];    System.out.println("Enter " + num + " integers: ");    for (i = 0; i < num; i++)  array[i] = x.nextInt();    for (i = 0; i < ( num - 1 ); i++) {  for (j = 0; j < num - i - 1; j++) {  if (array[j] > array[j+1])  {  temp = array[j];  array[j] = array[j+1];  array[j+1] = temp;  }  }  }    System.out.println("Sorted list of integers:");    for (i = 0; i < num; i++)  System.out.println(array[i]);    }  } |

**OUTPUT :**

Text, letter

Description automatically generated

**PROGRAM NO : 26**

**AIM:** Maintain a list of Strings using ArrayList from collection framework, perform built-in operations.

**ALGORITHM :**

STEP 1 : Start

STEP 2: Define an array .

STEP 3: Define objects to array.

STEP 4: Perform operation on array.

STEP 5: Print the array after operations.

STEP 6: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | package myproj;  import java.util.\*;  public class Q10 {  public static void main(String[] args) {  // TODO Auto-generated method stub  ArrayList<String> obj= new ArrayList<String>();  obj.add("Movies");  obj.add("Series");  obj.add("Documentaries");  obj.add("Animes");  //displaying array list  System.out.println("\nDisplaying Arraylist:\n");  for(String list:obj)  System.out.println(list);  //removing array elements from list  System.out.println("\nlist after Removing elements:\n");  obj.remove("Animes");  for(String list:obj)  System.out.println(list);  //sorting array list  System.out.println("\nSorting the list:\n");  Collections.sort(obj);  for(String list:obj)  System.out.println(list);    System.out.println("\nGetting object of list which is present at the specified index:\n");  System.out.println(obj.get(1));    //getting size of list  System.out.println("\nSize of the list:"+obj.size());    //clearing the array list  obj.clear();    System.out.println("\nArrayList after clear method :"+obj);    }  } |

**OUTPUT :**

Graphical user interface, text, application, email

Description automatically generated

**PROGRAM NO : 27**

**AIM:** Program to remove all the elements from a linked list

**ALGORITHM :**

STEP 1 : Start

STEP 2: Define a Linked lisr.

STEP 3: Define objects to Linked list using add().

STEP 4: Delete the elements in list using clear()

STEP 5: Print the Outputs.

STEP 6: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | package co4;  import java.util.\*;  public class linkedlist\_delete {  public static void main(String[] args) {    LinkedList<String> list = new LinkedList<String>();  list.add("anu");  list.add("manu");  list.add("rahul");  list.add("Robin");  list.add("hari");  list.add("Rani");    System.out.println("The Original linked list: " + list);  list.clear();  System.out.println("After removing : " + list);  }  } |

**OUTPUT :**

Text

Description automatically generated

**PROGRAM NO : 28**

**AIM:** Program to remove an object from the Stack when the position is passed as parameter.

**ALGORITHM :**

STEP 1 : Start

STEP 2: Define a Stack.

STEP 3: Define objects to Stack using add().

STEP 4:Enter key to remove the item.

STEP 5: Remove the item using remove(key).

STEP 6: Print outputs

STEP 7: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | //Program to remove an object from the Stack when the position is passed as parameter  package javaprograms;  import java.util.Scanner;  import java.util.Stack;  public class CO4Q12 {  public static void main(String[] args) {  int n;  String str;  Stack<String> s = new Stack<String>();  System.out.println("Enter the number of elements:");  Scanner sc = new Scanner(System.in);  n = sc.nextInt();  sc.nextLine();  System.out.println("Enter the elements:");  for(int i=0;i<n;i++)  {  str = sc.nextLine();  s.add(str);  }  System.out.println("\nStack elements:"+s);  System.out.println("\nTop element:"+s.peek());  System.out.println("Popped element:"+s.pop());  System.out.println("Stack elements after popped:"+s);  System.out.println("\nRemove Element at position 1:"+s.remove(0));  System.out.println("Stack elements after removed:"+s);  System.out.println("\nRemove Luke Skywalker:");  s.remove("Luke Skywalker");  System.out.println("Stack elements after removing Luke Skywalker:"+s);  }  } |

**OUTPUT :**

Graphical user interface, text, application, email

Description automatically generated

**PROGRAM NO : 29**

**AIM:** Program to demonstrate the creation of queue object using the PriorityQueue Class

**ALGORITHM :**

STEP 2: Define a Priority Queue.

STEP 3: Enter a limit n.

STEP 4: Add n elements to Queue.

STEP 5: Perform operations like remove(), add() etc.

STEP 6: Print outputs

STEP 7: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | //Program to demonstrate the creation of queue object using the PriorityQueue class  package javaprograms;  import java.util.Iterator;  import java.util.PriorityQueue;  import java.util.Scanner;  public class CO4Q13 {  public static void main(String[] args) {  int n;  String str;  PriorityQueue<String> pq = new PriorityQueue<String>();  System.out.println("Enter the no. of data:");  Scanner sc = new Scanner(System.in);  n = sc.nextInt();  sc.nextLine();  System.out.println("Enter the data:");  for(int i=0;i<n;i++)  {  str = sc.nextLine();  pq.add(str);  }  Iterator itr = pq.iterator();  System.out.println("\nPriority Queue\n");  while(itr.hasNext())  System.out.println(itr.next()+" ");  }  } |

**OUTPUT :**

Graphical user interface, application

Description automatically generated

**PROGRAM NO : 30**

**AIM:** Program to demonstrate the addition and deletion of elements in deque

**ALGORITHM :**

Step 1:Start

Step 2: Create an object of the class ArrayDeque.

Step 3: Adding elements to the queue using method add().

Step 4: Removing elements of queue using method pop().

Step 5: Display Queue.

Step 6: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | //Program to demonstrate the addition and deletion of elements in deque  package javaprograms;  import java.util.Deque;  import java.util.LinkedList;  import java.util.Scanner;  public class CO4Q14 {  public static void main(String[] args) {  int ch;  String data;  Deque<String> dq = new LinkedList<String>();  Scanner sc = new Scanner(System.in);  do  {  System.out.println("\nChoose a number...");  System.out.println("1.Insert the element at first");  System.out.println("2.Insert the element at last");  System.out.println("3.Delete the element at first");  System.out.println("4.Delete the element at last");  System.out.println("5.Display");  System.out.println("6.Exit");  System.out.println("\nEnter the choice:");  ch = sc.nextInt();  sc.nextLine();  switch(ch)  {  case 1: System.out.println("Enter the element to be inserted at first:");  data = sc.nextLine();  dq.addFirst(data);  break;  case 2: System.out.println("Enter the element to be inserted at last:");  data = sc.nextLine();  dq.addLast(data);  break;  case 3: System.out.println("Element deleted from the first position");  dq.removeFirst();  break;  case 4: System.out.println("Element deleted from the last position");  dq.removeLast();  break;  case 5: System.out.println("Elements:");  System.out.println(dq);  break;  case 6: System.exit(0);  break;  default:System.out.println("Invalid choice...");  }  }while(true);  }  } |

**OUTPUT**

**Text

Description automatically generated**

**Text

Description automatically generated with low confidence**

**Graphical user interface, text, application

Description automatically generated**

**PROGRAM NO : 31**

**AIM:** Program to demonstrate the creation of Set object using the LinkedHashset Class.

**ALGORITHM :**

STEP 1 : Start

STEP 2: Define a Linkedhash set.

STEP 3: Define objects to Linked hash set using add().

STEP 4:Perform Desired operations like adding already existing elements.

STEP 5: Remove the item using remove(key).

STEP 6: Print outputs

STEP 7: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | package c04;  import java.util.LinkedHashSet;  public class Q15 {  public static void main(String[] args) {  // TODO Auto-generated method stub  LinkedHashSet<String> linkedset =  new LinkedHashSet<String>();  // Adding elements to LinkedHashSet  linkedset.add("T");  linkedset.add("I");  linkedset.add("L");  linkedset.add("E");  //T will not be added as new elements,because it already exists  linkedset.add("T");  linkedset.add("S");  System.out.println("Size of LinkedHashSet = " +  linkedset.size());  System.out.println("Original LinkedHashSet:" + linkedset);  System.out.println("Removing E from LinkedHashSet: " +  linkedset.remove("E"));  System.out.println("Trying to Remove Z which is not "+  "present: " + linkedset.remove("Z"));  System.out.println("Checking if T is present=" +  linkedset.contains("T"));  System.out.println("Updated LinkedHashSet: " + linkedset);  }  } |

**OUTPUT :**

Text

Description automatically generated

**PROGRAM NO : 32**

**AIM:** Write a Java program to compare two hash sets.

**ALGORITHM :**

STEP 1 : Start

STEP 2: Define 2 hash sets.

STEP 3: Define objects to hash sets.

STEP 4: Compare the hash sets.

STEP 5: Print the output of s1 union s2

STEP 6: Print output of s1 intersection s2

STEP 7: Print output of s1 difference s2

STEP 8: Print output of s2 difference s1

STEP 9: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | package c04;  import java.util.HashSet;  import java.util.Set;  public class Q16 {  public static void main(String[] args) {  // Creating the sets  Set<String> s1 = new HashSet<>();  s1.add("CSS");  s1.add("HTML");  s1.add("Java");  Set<String> s2 = new HashSet<>();  s2.add("CSS");  s2.add("Javascript");  s2.add("SQL");  //Printing elements of both the hashsets  System.out.println("Elements of first Hash set (s1) :" + s1);  System.out.println("Elements of second Hash set (s2) : " + s2);  // Perform set operations  performUnion(s1, s2);  performIntersection(s1, s2);  performDifference(s1, s2);  }  public static void performUnion(Set<String> s1, Set<String> s2) {  Set<String> s1Unions2 = new HashSet<>(s1);  s1Unions2.addAll(s2);  System.out.println("s1 union s2: " + s1Unions2);  }  public static void performIntersection(Set<String> s1, Set<String> s2) {  Set<String> s1Intersections2 = new HashSet<>(s1);  s1Intersections2.retainAll(s2);  System.out.println("s1 intersection s2: " + s1Intersections2);  }  public static void performDifference(Set<String> s1, Set<String> s2) {  Set<String> s1Differences2 = new HashSet<>(s1);  s1Differences2.removeAll(s2);  Set<String> s2Differences1 = new HashSet<>(s2);  s2Differences1.removeAll(s1);  System.out.println("s1 difference s2: " + s1Differences2);  System.out.println("s2 difference s1: " + s2Differences1);  }  } |

**OUTPUT :**

Text

Description automatically generated

**PROGRAM NO : 33**

**AIM:** Program to demonstrate the working of Map interface by adding, changing and removing elements.

**SOLUTION:** In Eclipse IDE type the necessary code and save with the file name mentioned with java extension.

**ALGORITHM :**

STEP 1 : Start

STEP 2: Define a Map set mp.

STEP 3: Define objects to Map set mp using put().

STEP 4: Remove the elements using remove(); .

STEP 5: Print the Outputs.

STEP 6: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.\*;  public class co4\_q17 {  public static void main(String args[])  {  Map<Integer, String> mp = new HashMap<>();  //Inserting elements..  mp.put(1, "TONY");  mp.put(2, "STEVE");  mp.put(3, "THOR");  System.out.println("Initial Map:"+mp);  mp.put( (2), "HAWKEYE");  mp.put((4), "NATASHA");  //Updating..  System.out.println("Updated Map:"+mp);  //Removing..  mp.remove(4);  // Final Map..  System.out.println("After Removing 4th entry, Final Map is:"+mp);  }  } |

**OUTPUT :**

Text

Description automatically generated

**PROGRAM NO : 34**

**AIM:** Program to Convert HashMap to TreeMap.

**SOLUTION:** In Eclipse IDE type the necessary code and save with the file name mentioned with java extension.

**ALGORITHM :**

STEP 1 : Start

STEP 2: Define a map mp.

STEP 3: Define objects to mp using put().

STEP 4: Define a tree set treeMap

STEP 4:Convert map set into tree using treeMap.putAll(map).

STEP 5: Print the Outputs.

STEP 6: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.\*;  public class co4\_q18 {  public static void main(String args[]) {  Map<String, String> map = new HashMap<>();  map.put("1","kollam");  map.put("2","tvm");  map.put("3","alpy");    System.out.println("HashMap:"+map);  Map<String, String> treeMap = new TreeMap<>();  treeMap.putAll(map);  System.out.println("TreeMap:"+treeMap);  }  } |

**OUTPUT :**

**Text

Description automatically generated**