PROGRAM NO : 1

AIM: Define a class 'product' with data members pcode, pname and price. Create 3 objects of the class and find the product having the lowest price.

ALGORITHM:

Step 1: Start.

Step 2: Define a class having name Product and members as pcode,pname and price.

Step 3: Declare three objects in the class and add the values of each data members into objects.

Step 4: Using if condition check which object has the lowest price and print it.

Step 5: Stop.

PROGRAM CODE:

Main.java	
| | ```
import java.util.Scanner;

public class Main {

   public static void main(String[] args) {
      //Driver program

      Scanner scan = new Scanner(System.in);
      Product[] products = new Product[3];

      for(int i=0;i<3;i++)
      {
         int count = i+1;

         products[i] = new Product();
         System.out.print("Enter product "+count+" name : ");
         products[i].pname = scan.nextLine();

         System.out.print("Enter product "+count+" code : ");
         products[i].pcode = scan.nextLine();

         System.out.print("Enter product "+count+" price : ");
         products[i].price = Integer.parseInt(scan.nextLine());
``` |

```
        }

        scan.close();

        if(products[0].price< products[1].price &&  products[0].price<
products[2].price) {
            System.out.println("The product with lowest price is:"+
products[0].pname);
        }
        if(products[1].price<products[0].price &&
products[1].price<products[2].price) {
            System.out.println("The product with lowest price
is:"+products[1].pname);
        }
        if(products[2].price<products[0].price &&
products[2].price<products[1].price) {
            System.out.println("The product with lowest price
is:"+products[2].pname);
        }


    }
}
package com.lab_cycles.co1.q1;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        //Driver program

        Scanner scan = new Scanner(System.in);
        Product[] products = new Product[3];

        for(int i=0;i<3;i++)
        {
            int count = i+1;

            products[i] = new Product();
            System.out.print("Enter product "+count+" name : ");
            products[i].pname = scan.nextLine();

            System.out.print("Enter product "+count+" code : ");
            products[i].pcode = scan.nextLine();
```

	System.out.print("Enter product "+count+" price : ");
	products[i].price = Integer.parseInt(scan.nextLine());
	
	}
	
	scan.close();
	
	if(products[0].price< products[1].price && products[0].price<
	products[2].price) {
	System.out.println("The product with lowest price is:"+
	products[0].pname);
	}
	if(products[1].price<products[0].price &&
	products[1].price<products[2].price) {
	System.out.println("The product with lowest price
	is:"+products[1].pname);
	}
	if(products[2].price<products[0].price &&
	products[2].price<products[1].price) {
	System.out.println("The product with lowest price
	is:"+products[2].pname);
	}
	
	    }
	}

Product.java	public class Product {
	    public String pcode;
	    public String pname;
	    public int price;
	}

RESULT: The above program is executed and obtained the output.

OUTPUT:

```
Run:    Main (1)
    C:\Users\Razik\.jdks\corretto-1.8.0_292\bin\java.exe ...
    Enter product 1 name : Jam
    Enter product 1 code : j
    Enter product 1 price : 70
    Enter product 2 name : Butter
    Enter product 2 code : b
    Enter product 2 price : 150
    Enter product 3 name : Bread
    Enter product 3 code : b
    Enter product 3 price : 40
    The product with lowest price is:Bread

    Process finished with exit code 0
```

PROGRAM NO : 2

AIM: Read 2 matrices from the console and perform matrix addition.

ALGORITHM :

Step 1: Start.

Step 2: Define a class having name AddMatrix.

Step 3: Read row number(m),column number (n) and initialize the double dimensional arrays mat1[][],mat2[][],res[][] with same row number ,column number.

Step 4: Store the first matrix elements into the two-dimensional array matrix mat1[][] using two for loops. i indicates row number, j indicates column index. Similarly second matrix elements in to mat2[][].

Step 5: Add the two matrices using for loop.
> for i=0 to i<m
>
> for j=0 to j<n
>
> mat1[i][j] + mat2[i][j] and store it in to the matrix res[i][j] .

Step 6: Print sum of matrices res[i][j].

Stop 7: Stop.

PROGRAM CODE:

AddMatrix .java	import java.util.Scanner;
	
	 public class AddMatrix {
	
	   public static void main(String args[])
	   {
	     int row, col,i,j;
	     Scanner in = new Scanner(System.in);
	
	     System.out.println("Enter the number of rows : ");
	     row = in.nextInt();
	
	     System.out.println("Enter the number columns : ");
	     col = in.nextInt();
	
	     int mat1[][] = new int[row][col];

```
        int mat2[][] = new int[row][col];
        int res[][] = new int[row][col];

        System.out.println("Enter the elements of matrix 1  : ");

        for ( i= 0 ; i < row ; i++ )
        {

           for ( j= 0 ; j < col ;j++ )
              mat1[i][j] = in.nextInt();

           System.out.println();
        }
        System.out.println("Enter the elements of matrix 2  : ");

        for ( i= 0 ; i < row ; i++ )
        {

           for ( j= 0 ; j < col ;j++ )
              mat2[i][j] = in.nextInt();

           System.out.println();
        }

        for ( i= 0 ; i < row ; i++ )
           for ( j= 0 ; j < col ;j++ )
              res[i][j] = mat1[i][j] + mat2[i][j] ;
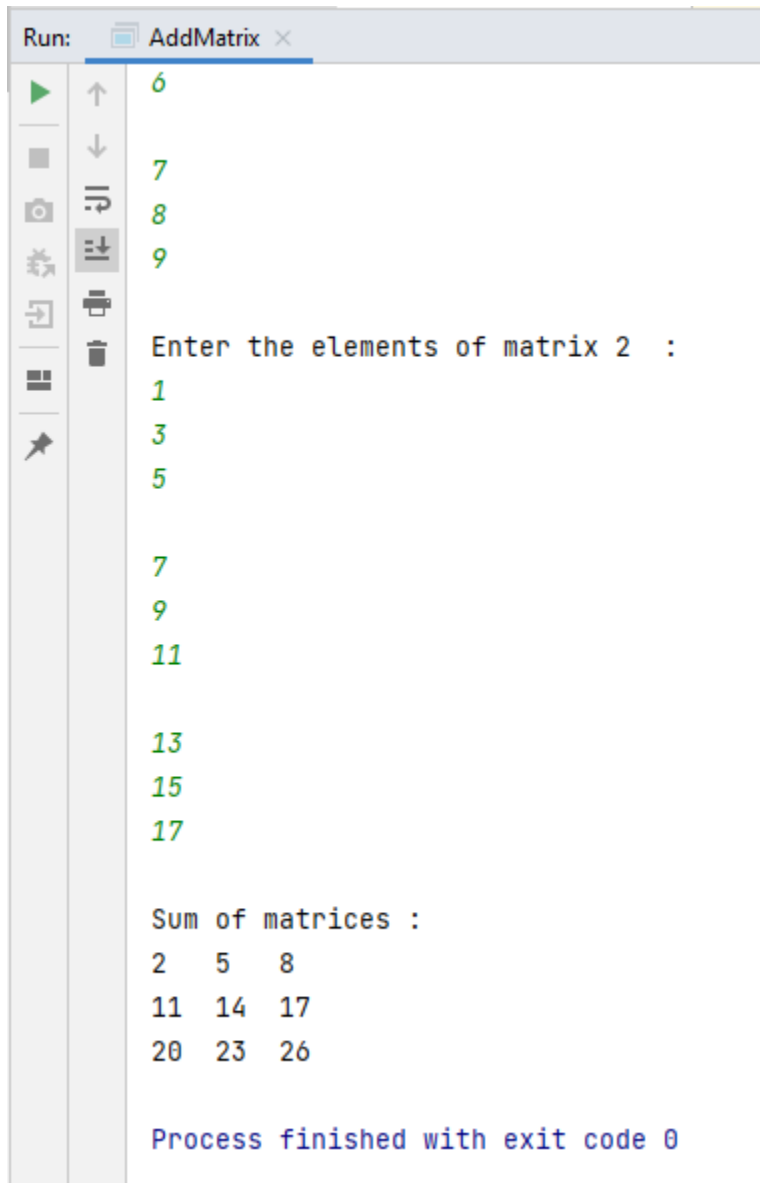
        System.out.println("Sum of matrices : ");

        for ( i= 0 ; i < row ; i++ )
        {
           for ( j= 0 ; j < col ;j++ )
              System.out.print(res[i][j]+"\t");

           System.out.println();
        }

     }

}
```

RESULT: The above program is executed and obtained the output.

OUTPUT:

```
Run:        AddMatrix ×
    6

    7
    8
    9

    Enter the elements of matrix 2   :
    1
    3
    5


    7
    9
    11


    13
    15
    17


    Sum of matrices :
    2    5    8
    11   14   17
    20   23   26

    Process finished with exit code 0
```

PROGRAM NO : 3

AIM: Add complex numbers.

ALGORITHM:

Step 1: Start.

Step 2: Define a class having name ComplexNumber and data members are real and imaginary number.

Step 3: Define a function ComplexNumber and add values to variables.

Step 4 : Define a function ComplexNumber sum to add complex number using 3rd ComplexNumber object and return the value.

Step 5: Print the sum value.

Step 6: Stop.

PROGRAM CODE:

ComplexNumber. java	public class ComplexNumber {
	double real,imaginary;
	ComplexNumber(double real,double imaginary)
	{
	this.real = real;
	this.imaginary = imaginary;
	}
	ComplexNumber addComplexNumber(ComplexNumber complexNumber1,ComplexNumber complexNumber2)
	{
	double real = complexNumber1.real+ complexNumber2.real;

7

| | ```java
        double imaginary =
complexNumber1.imaginary+complexNumber2.imaginary;

        ComplexNumber complexNumber = new
ComplexNumber(real,imaginary);

        return complexNumber;

   }

   String displayComplexNumber()

   {

      return this.real+" + "+this.imaginary+"i";

   }

}
``` |
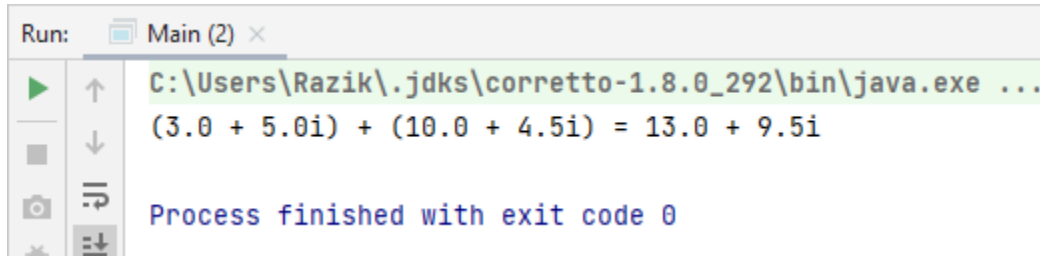| Main.java | ```java
public class Main {

   public static void main(String[] args) {

      ComplexNumber complexNumber1 = new ComplexNumber(3,5);

      ComplexNumber complexNumber2 = new ComplexNumber(10,4.5);

      ComplexNumber complexNumber3 = new ComplexNumber(0,0);

      complexNumber3 =
complexNumber3.addComplexNumber(complexNumber1,complexNumbe
r2);



System.out.println("("+complexNumber1.displayComplexNumber()+") +
("+complexNumber2.displayComplexNumber()+") =
"+complexNumber3.displayComplexNumber());

   }

}
``` |

RESULT: The above program is executed and obtained the output.

OUTPUT:

```
Run:    Main (2) ×
    ↑   C:\Users\Razik\.jdks\corretto-1.8.0_292\bin\java.exe ...
    ↓   (3.0 + 5.0i) + (10.0 + 4.5i) = 13.0 + 9.5i

        Process finished with exit code 0
```

PROGRAM NO : 4

AIM: Read a matrix from the console and check whether it is symmetric or not.

ALGORITHM:

Step 1: Start.

Step 2 : Read row number,column number and initialize the double dimensional array with same row number ,column number.

Step 3 : Store the first matrix elements into the two-dimensional array matrix using two for loops. i indicates row number, j indicates column index.

Step 4: Check whether the matrix is symmetric or not.

Step 5: Print the symmetric matrix or if not.

Step 6: Stop.

PROGRAM CODE:

Main.java	

	import java.util.Scanner;
	
	 public class Main
	
	 {
	
	    public static void main(String[] args)
	
	    {
	
	       Scanner sc = new Scanner(System.in);
	
	       System.out.println("Enter the no. of rows : ");
	
	       int rows = sc.nextInt();
	
	       System.out.println("Enter the no. of columns : ")

```
        int cols = sc.nextInt();

        int matrix[][] = new int[rows][cols];

        System.out.println("Enter the elements :");

        for (int i = 0; i < rows; i++)

        {

            for (int j = 0; j < cols; j++)

            {

                System.out.println("Input number for matrix position
["+(i+1)+"]["+(j+1)+"] : ");

                matrix[i][j] = sc.nextInt();

            }

        }


        System.out.println("Printing the input matrix :");

        for (int i = 0; i < rows; i++)

        {

            for (int j = 0; j < cols; j++)

            {

                System.out.print(matrix[i][j]+"\t");

            }

            System.out.println();

        }
```

```java
//Checking the input matrix for symmetric

if(rows != cols)

{

    System.out.println("The given matrix is not a square matrix, so it
can't be symmetric.");

}

else

{

    boolean symmetric = true;


    for (int i = 0; i < rows; i++)

    {

        for (int j = 0; j < cols; j++)

        {

            if(matrix[i][j] != matrix[j][i])

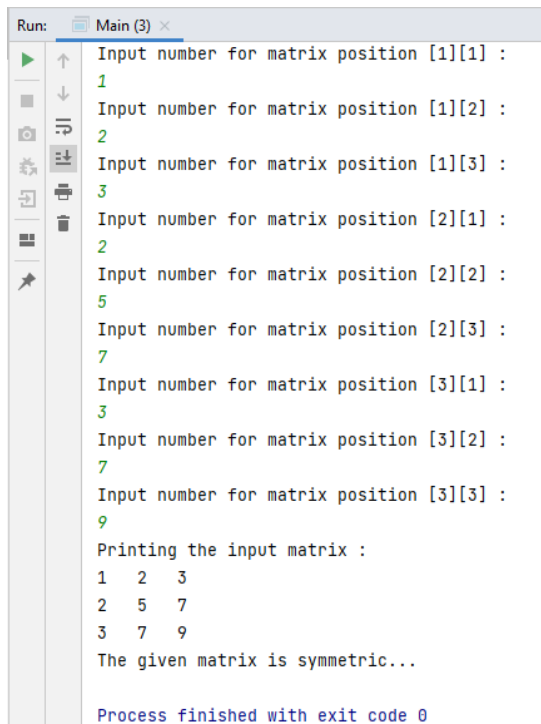            {

                symmetric = false;

                break;

            }

        }

    }


    if(symmetric)
```

	{
	
	System.out.println("The given matrix is symmetric...");
	
	}
	
	else
	
	{
	
	System.out.println("The given matrix is not symmetric...");
	
	}
	
	}
	
	sc.close();
	
	}
	
	}

RESULT: The above program is executed and obtained the output.

OUTPUT:

```
Run:    Main (3) ×
    Input number for matrix position [1][1] :
    1
    Input number for matrix position [1][2] :
    2
    Input number for matrix position [1][3] :
    3
    Input number for matrix position [2][1] :
    2
    Input number for matrix position [2][2] :
    5
    Input number for matrix position [2][3] :
    7
    Input number for matrix position [3][1] :
    3
    Input number for matrix position [3][2] :
    7
    Input number for matrix position [3][3] :
    9
    Printing the input matrix :
    1   2   3
    2   5   7
    3   7   9
    The given matrix is symmetric...

    Process finished with exit code 0
```

PROGRAM NO : 5

AIM: Create CPU with attribute price. Create inner class Processor (no. of cores, manufacturer) and static nested class RAM (memory, manufacturer). Create an object of CPU and print information of Processor and RAM.

ALGORITHM :

Step 1: Start.

Step 2: Define a class cpu with data member price and class processor.

Step 3: Class processor contain data members no_cores,manufacturer and a nested class RAM.

Step 4: class RAM contain memory and manufacturer as data members.

Step 5: Create objects in corresponding classes and display it's details.

Step 6: Stop.

PROGRAM CODE:

| CPU.java | ```java
public class CPU {

    int price;
    Processor processor = new Processor();
    RAM ram = new RAM();

    static class Processor
    {
        int cores;
        String manufacturer;
    }

    static class RAM
    {
        int memory;
        String manufacturer;
    }
``` |
|---|---|

	CPU(int cpuPrice,int cores,String cpuManufacturer,int memory,String ramManufacturer){
	    this.price = cpuPrice;
	    this.processor.cores = cores;
	    this.processor.manufacturer = cpuManufacturer;
	    this.ram.memory = memory;
	    this.ram.manufacturer = ramManufacturer;
	  }
	
	  void printCPUSpecifications()
	  {
	    System.out.println("CPU Specifications");
	    System.out.println("---");
	    System.out.println("CPU price : "+this.price);
	    System.out.println("Processor cores : "+this.processor.cores);
	    System.out.println("Processor manufacturer : "+this.processor.manufacturer);
	    System.out.println("RAM memory : "+this.ram.memory);
	    System.out.println("RAM manufacturer : "+this.ram.manufacturer);
	    System.out.println("---");
	  }
	
	}

Main.java	public class Main {
	
	  public static void main(String[] args) {
	
	    CPU cpu1 = new CPU(50000, 8, "INTEL", 8000, "ADATA");
	    cpu1.printCPUSpecifications();
	  }
	
	}

RESULT: The above program is executed and obtained the output.

OUTPUT:

PROGRAM NO : 6

AIM: Program to Sort strings.

ALGORITHM :

Step 1: Start

Step 2: Select the first element of the list (i.e., Element at first position in the list).

Step 3: Compare the selected element with all the other elements in the list.

Step 4: In every comparision, if any element is found smaller than the selected element (for Ascending order), then both are swapped.

Step 5: Repeat the same procedure with element in the next position in the list till the entire list is sorted.

Step 6: Stop

.

PROGRAM CODE:

| Sort.java | import java.util.Scanner;

public class Sort {

  public static void main(String[] args) {

    int count;
    String temp;
    Scanner scan = new Scanner(System.in);

    //User will be asked to enter the count of strings
    System.out.print("Enter number of strings you would like to enter: ");
    count = scan.nextInt();

    String[] str = new String[count];
    Scanner scan2 = new Scanner(System.in); |

```
            //User is entering the strings and they are stored in an array
            System.out.println("Enter the Strings one by one: ");
            for(int i = 0; i < count; i++)
            {
               str[i] = scan2.nextLine();
            }
            scan.close();
            scan2.close();

            //Sorting the strings
            for (int i = 0; i < count; i++)
            {
               for (int j = i + 1; j < count; j++) {
                  if (str[i].compareTo(str[j])>0)
                  {
                     temp = str[i];
                     str[i] = str[j];
                     str[j] = temp;
                  }
               }
            }

            //Displaying the strings after sorting them based on alphabetical order
            System.out.print("Strings in Sorted Order: ");
            for (int i = 0; i <= count - 1; i++)
            {
               System.out.print(str[i] + ", ");
            }

         }

   }
```
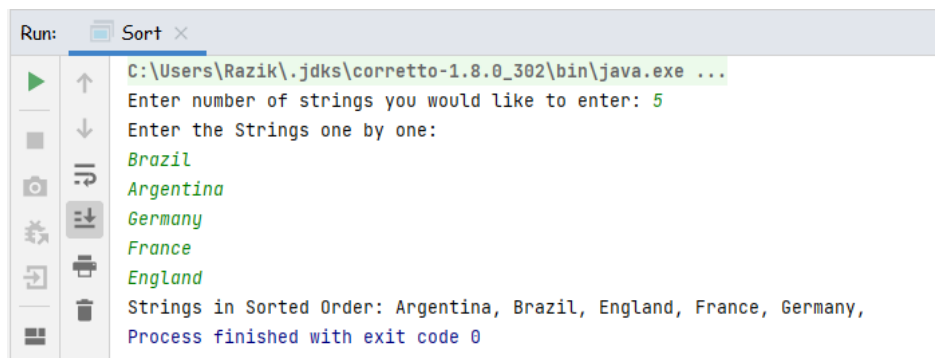
RESULT: The above program is executed and obtained the output.

OUTPUT:


```
Run:      Sort ×
          C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
          Enter number of strings you would like to enter: 5
          Enter the Strings one by one:
          Brazil
          Argentina
          Germany
          France
          England
          Strings in Sorted Order: Argentina, Brazil, England, France, Germany,
          Process finished with exit code 0
```

PROGRAM NO : 7

AIM: Search an element in an array.

ALGORITHM :

Step 1: Start

Step 2: Check each element in the given list with the string provided by the user.

Step 3: If string is found, display the position of the string found, else display string not found.

Step : Stop

PROGRAM CODE:

SearchItem.java	
| | ```java
import java.util.Scanner;

public class SearchItem {

  public static void main(String[] args) {

    int count;
    String temp;
    Scanner scan = new Scanner(System.in);
    Scanner scan2 = new Scanner(System.in);
    Scanner scan3 = new Scanner(System.in);

    //User will be asked to enter the count of strings
    System.out.print("Enter number of strings you would like to enter: ");
    count = scan.nextInt();


    String[] str = new String[count];

    //User is entering the strings and they are stored in an array
    System.out.println("Enter the Strings one by one: ");
    for(int i = 0; i < count; i++)
    {
       str[i] = scan2.nextLine();
    }


    System.out.println("Enter the String to be searched : ");
``` |

```java
        String searchString = scan3.nextLine();


        for (int i = 0; i < count; i++)
        {
            if(str[i].equals(searchString)){
                System.out.println("The required string is in position : "+(i));
                System.exit(0);
            }
        }

        scan.close();
        scan2.close();
        scan3.close();

        System.out.println("Could not find required string in Array.");

    }

}
```

RESULT: The above program is executed and obtained the output.

OUTPUT:

```
Run:    SearchItem ×

    C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
    Enter number of strings you would like to enter: 8
    Enter the Strings one by one:
    Blue
    Red
    Green
    Purple
    Violet
    Orange
    Grey
    Gold
    Enter the String to be searched :
    Violet
    The required string is in position : 4

    Process finished with exit code 0
```

PROGRAM NO : 8

AIM: Perform string manipulations

ALGORITHM :

Step 1: Start

Step 2: Take the strings provided by the user and concatenate them.

Step 3: Display the combined string with lower case.

Step 3: Display the combined string with upper case.

Step 4: Display the combined string after replacing all the 's' & 'S' characters with '$' character.

Step 5: Stop

PROGRAM CODE:

StringMan.java	
| | ```java
import java.util.Scanner;

public class SearchItem {

    public static void main(String[] args) {

        int count;
        String temp;
        Scanner scan = new Scanner(System.in);
        Scanner scan2 = new Scanner(System.in);
        Scanner scan3 = new Scanner(System.in);

        //User will be asked to enter the count of strings
        System.out.print("Enter number of strings you would like to enter: ");
        count = scan.nextInt();


        String[] str = new String[count];

        //User is entering the strings and they are stored in an array
        System.out.println("Enter the Strings one by one: ");
        for(int i = 0; i < count; i++)
        {
``` |

```java
            str[i] = scan2.nextLine();
        }


        System.out.println("Enter the String to be searched : ");
        String searchString = scan3.nextLine();


        for (int i = 0; i < count; i++)
        {
           if(str[i].equals(searchString)){
               System.out.println("The required string is in position : "+(i));
               System.exit(0);
           }
        }

        scan.close();
        scan2.close();
        scan3.close();

        System.out.println("Could not find required string in Array.");

    }
}
```

RESULT: The above program is executed and obtained the output.

OUTPUT:

PROGRAM NO : 9

AIM: Program to create a class for Employee having attributes eNo, eName eSalary. Read n employ information and Search for an employee given eNo, using the concept of Array of Objects.

ALGORITHM :

Step 1: Start

Step 2: Search the 'eNo' attribute of the list of Employee Objects for the 'eNo' provided by the user.

Step 3: If user provided 'eNo' is found inside the Employee object list, display the details of the corresponding employee.

Step 4: Stop

PROGRAM CODE:

| Employee.java | ```java
public class Employee {

    public String eNo;
    public String eName;
    public int eSalary;

    public Employee(String eNo,String eName,int eSalary)
    {
       this.eNo = eNo;
       this.eName = eName;
       this.eSalary = eSalary;
    }

}
``` |
| Main.java | ```java
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
``` |

```
        Scanner scan = new Scanner(System.in);

        System.out.print("Enter count of employee list : ");
        int count = Integer.parseInt(scan.nextLine());

        Employee[] employees = new Employee[count];

        for (int i=0;i<count;i++)
        {
          System.out.print("Enter Number of employee : ");
          String eNo = scan.nextLine();
          System.out.print("Enter Name of employee : ");
          String eName = scan.nextLine();
          System.out.print("Enter Salary of employee : ");
          int eSalary = Integer.parseInt(scan.nextLine());
          employees[i] = new Employee(eNo,eName,eSalary);
        }

        System.out.print("Enter the number of employee to be filtered out :
");
        String search = scan.nextLine();

        for (int i=0;i<count;i++){

          if(employees[i].eNo.equals(search))
          {
            System.out.println("Employee Number : "+employees[i].eNo);
            System.out.println("Employee Name : "+employees[i].eName);
            System.out.println("Employee Salary : "+employees[i].eSalary);
          }

        }

    }
}
```
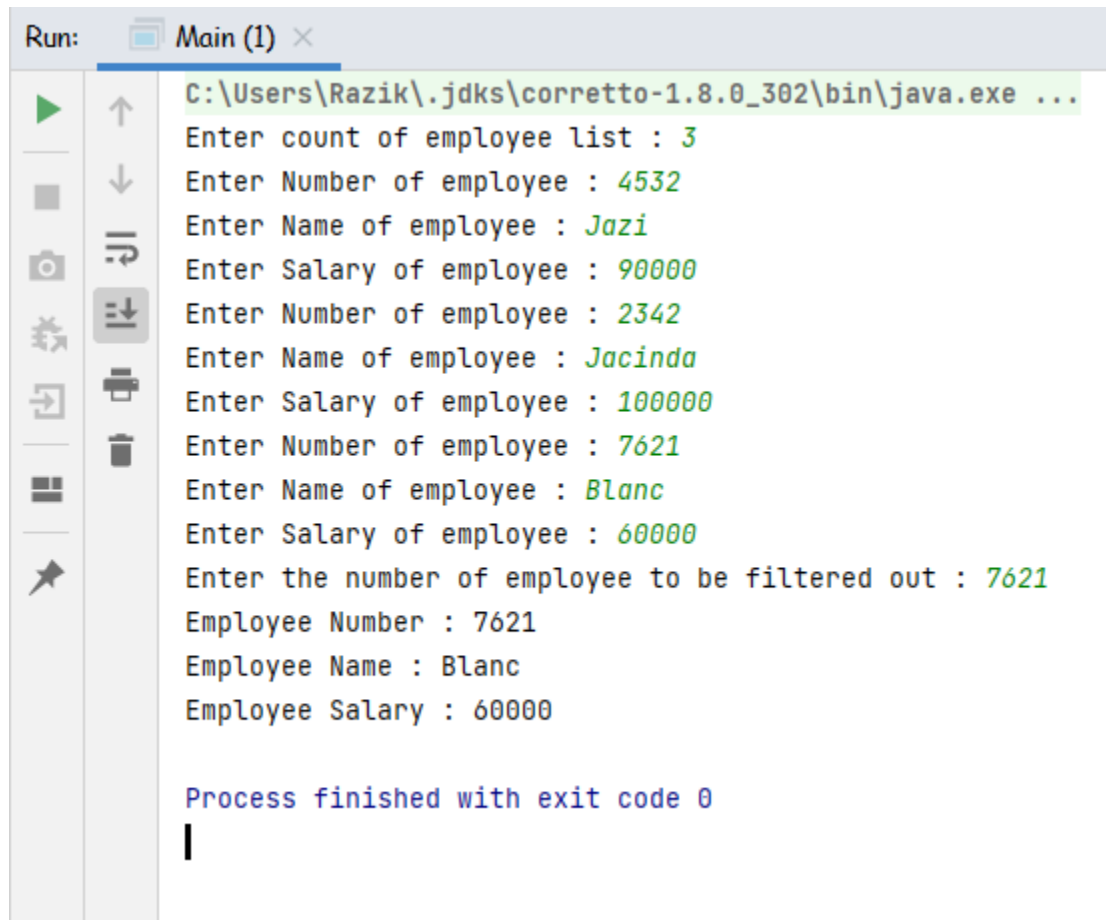
RESULT: The above program is executed and obtained the output.

OUTPUT:

```
Run:    Main (1) ×

    C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
    Enter count of employee list : 3
    Enter Number of employee : 4532
    Enter Name of employee : Jazi
    Enter Salary of employee : 90000
    Enter Number of employee : 2342
    Enter Name of employee : Jacinda
    Enter Salary of employee : 100000
    Enter Number of employee : 7621
    Enter Name of employee : Blanc
    Enter Salary of employee : 60000
    Enter the number of employee to be filtered out : 7621
    Employee Number : 7621
    Employee Name : Blanc
    Employee Salary : 60000

    Process finished with exit code 0
```

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 | ```java
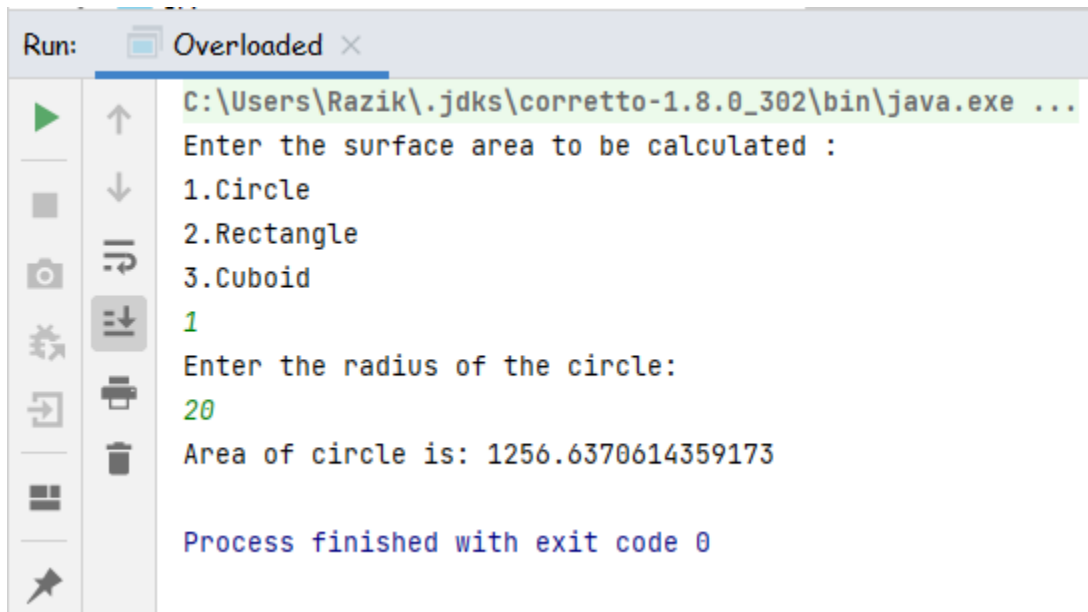public class Areas {
    public void area(float length,float breadth)
    {
        System.out.println("Area of reactangle is: "+length*breadth);
    }

    public void area(float radius)
    {
        System.out.println("Area of circle is: "+(radius*radius*Math.PI));
    }

    public void area(float length,float breadth, float height)
    {
        float area = 2*((length*breadth)+(breadth*height)+(length*height));
        System.out.println("Area of cuboid is: "+area);
    }
}
``` |
|---|---|
| Overloaded.java | ```java
import java.util.Scanner;

public class Overloaded {

    public static void main(String[] args) {
        float a,b,c;
        int choice;

        Scanner scanner=new Scanner(System.in);
``` |

25

```
                      Areas areas =  new Areas();

                      System.out.println("Enter the surface area to be calculated : ");
                      System.out.println("1.Circle");
                      System.out.println("2.Rectangle");
                      System.out.println("3.Cuboid");
                      choice = scanner.nextInt();

                      if(choice==1)
                      {
                        System.out.println("Enter the radius of the circle: ");
                        a=scanner.nextInt();
                        areas.area(a);
                      }
                      else if(choice==2)
                      {
                        System.out.println("Enter the length of the rectangle: ");
                        a=scanner.nextInt();
                        System.out.println("Enter the breadth of the rectangle: ");
                        b=scanner.nextInt();
                        areas.area(a,b);
                      }
                      else if(choice==3)
                      {
                        System.out.println("Enter the length of the cuboid: ");
                        a=scanner.nextInt();
                        System.out.println("Enter the breadth of the cuboid: ");
                        b=scanner.nextInt();
                        System.out.println("Enter the height of the cuboid: ");
                        c=scanner.nextInt();
                        areas.area(a,b,c);
                      }
                      else
                      {
                        System.out.println("Invalid Choice");
                      }
                  }
              }
```
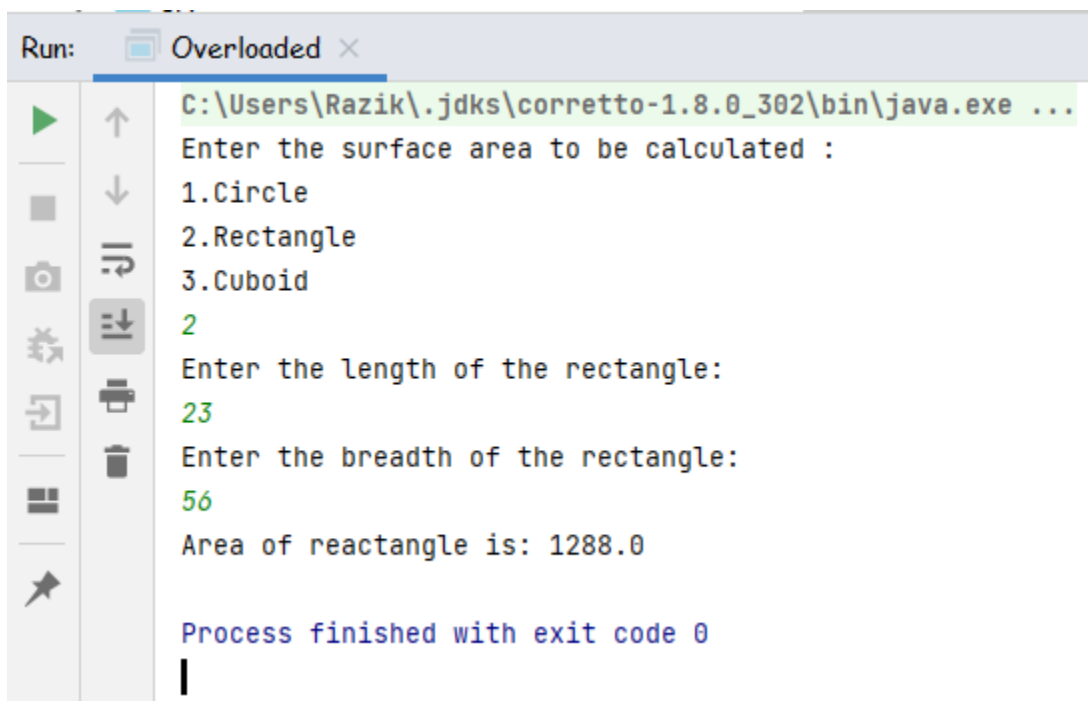
RESULT: The above program is executed and obtained the output.

OUTPUT:

```
Run:    Overloaded ×
        C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
        Enter the surface area to be calculated :
        1.Circle
        2.Rectangle
        3.Cuboid
        1
        Enter the radius of the circle:
        20
        Area of circle is: 1256.6370614359173

        Process finished with exit code 0
```

```
Run:    Overloaded ×
        C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
        Enter the surface area to be calculated :
        1.Circle
        2.Rectangle
        3.Cuboid
        2
        Enter the length of the rectangle:
        23
        Enter the breadth of the rectangle:
        56
        Area of reactangle is: 1288.0

        Process finished with exit code 0
```

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	```java
	public class Employee {
	  String Empid;
	  String Name;
	  String Salary;
	  String Address;
	
	  Employee(String Empid,String Name,String Salary,String Address)
	  {
	    this.Empid = Empid;
	    this.Name = Name;
	    this.Salary = Salary;
	    this.Address = Address;
	  }
	
	}
	``

Teacher.java	```java
	public class Teacher extends Employee {
	
	    String department;
	    String Subjects;
	``

| | ```java
Teacher(String Empid, String Name, String Salary, String
Address,String department,String Subjects) {
    super(Empid, Name, Salary, Address);
    this.department = department;
    this.Subjects = Subjects;
}

void displayTeacherDetails()
{
    System.out.println("Empid : "+this.Empid+" Name : "+this.Name+"
Salary : "+this.Salary+" Address : "+this.Address+"department :
"+this.department+" Subjects : "+this.Subjects);
}

}
``` |
| Main.java | ```java
import java.util.Scanner;

public class Main {

    public static void main(String[] args)
    {
        int count;
        Scanner scanner = new Scanner(System.in);
        System.out.print("Input the number of teachers required : ");
        count = scanner.nextInt();
        Teacher[] teachers = new Teacher[count];

        for(int i=0;i<count;i++) {
            int j = i+1;
            System.out.print("Enter Empid of teacher "+j+" : ");
            String Empid = scanner.next();
            System.out.print("Enter Name of teacher "+j+" : ");
            String Name = scanner.next();
            System.out.print("Enter Salary of teacher "+j+" : ");
            String Salary = scanner.next();
            System.out.print("Enter Address of teacher "+j+" : ");
            String Address = scanner.next();
            System.out.print("Enter department of teacher "+j+" : ");
            String department = scanner.next();
            System.out.print("Enter Subjects of teacher "+j+" : ");
            String Subjects = scanner.next();

            teachers[i] = new Teacher(Empid, Name, Salary, Address,
department, Subjects);
        }
``` |
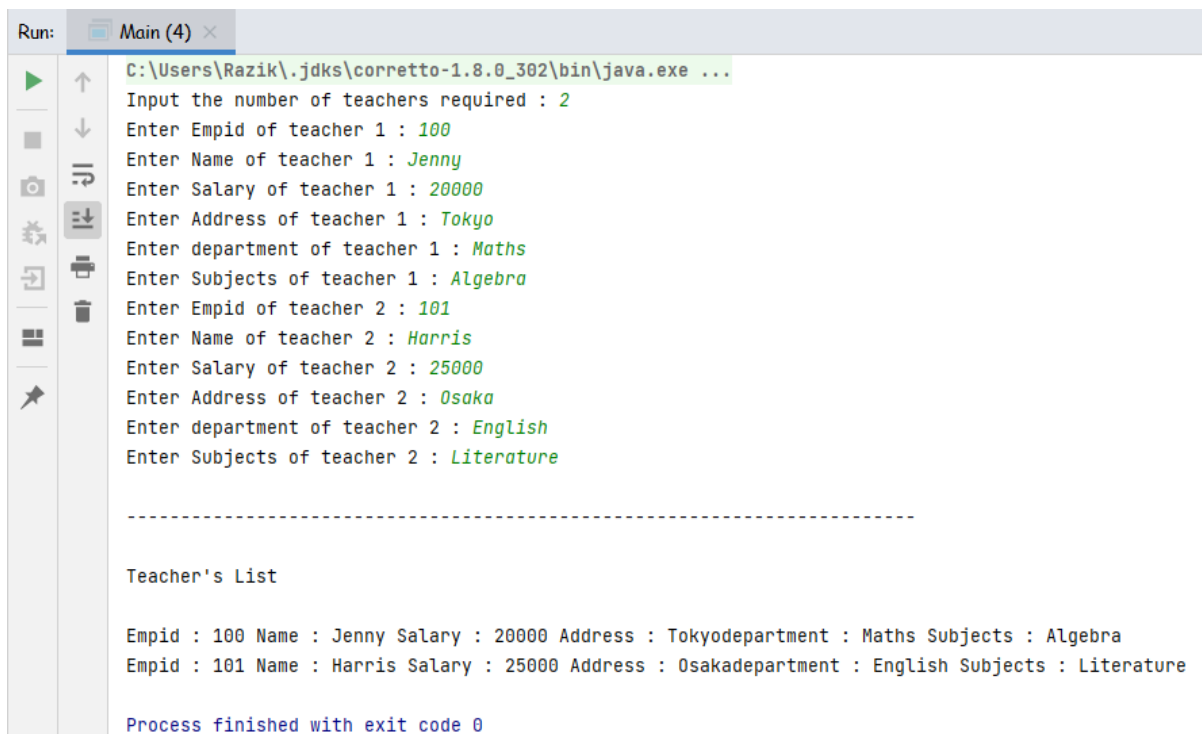
```
            System.out.println("\n----------------------------------------------------------
----------------\n");
            System.out.println("Teacher's List \n");


            for(int i=0;i<count;i++) {
                teachers[i].displayTeacherDetails();
            }

        }

    }
```

RESULT: The above program is executed and obtained the output.

OUTPUT:

```
Run:        Main (4) ×
            C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
            Input the number of teachers required : 2
            Enter Empid of teacher 1 : 100
            Enter Name of teacher 1 : Jenny
            Enter Salary of teacher 1 : 20000
            Enter Address of teacher 1 : Tokyo
            Enter department of teacher 1 : Maths
            Enter Subjects of teacher 1 : Algebra
            Enter Empid of teacher 2 : 101
            Enter Name of teacher 2 : Harris
            Enter Salary of teacher 2 : 25000
            Enter Address of teacher 2 : Osaka
            Enter department of teacher 2 : English
            Enter Subjects of teacher 2 : Literature


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

            Teacher's List

            Empid : 100 Name : Jenny Salary : 20000 Address : Tokyodepartment : Maths Subjects : Algebra
            Empid : 101 Name : Harris Salary : 25000 Address : Osakadepartment : English Subjects : Literature

            Process finished with exit code 0
```

PROGRAM NO : 12

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

ALGORITHM :

Step 1: Start

Step 2: Create a class named 'Person' with data members name, gender, address and age

& a constructor to initialize them.

Step 3: Create a class named 'Employee' which is derived from Person, with data members

empid, cmpnyname, qualification and sal & a constructor Employee() to initialize them.

Step 4: Create class named 'Teach" which is derived from Employee, with data members

subject, dept and tid ; a constructor to initilize members ; and a function named

display() to display details.

Step 5: Create an array of objects to display details.

Step 6: Stop

PROGRAM CODE:

| Employee.java | ```
public class Employee extends Person{
  String Empid;
  String Salary;

  Employee(String Name,String Gender,String Address,String Age,String
Empid,String Salary)
  {
    super(Name,Gender,Address,Age);
    this.Empid = Empid;
    this.Salary = Salary;
  }

}
``` |
|---|---|
| Teacher.java | ```
public class Teacher extends Employee {

    String department;
    String Subjects;
``` |

31

| | ```
Teacher(String Name,String Gender,String Address,String Age,String
Empid,String Salary,String department,String Subjects) {
    super(Name,Gender,Address,Age,Empid,Salary);
    this.department = department;
    this.Subjects = Subjects;
}

void displayTeacherDetails()
{
    System.out.println("Empid : "+this.Empid+" Name : "+this.Name+"
Gender : "+this.Gender+" Age : "+this.Age+" Salary : "+this.Salary+"
Address : "+this.Address+"department : "+this.department+" Subjects :
"+this.Subjects);
}

}        System.out.println("Empid : "+this.Empid+" Name : "+this.Name+"
Salary : "+this.Salary+" Address : "+this.Address+"department :
"+this.department+" Subjects : "+this.Subjects);
}

}
``` |
| Person.java | ```
public class Person {

    String Name;
    String Gender;
    String Address;
    String Age;

    Person(String Name,String Gender,String Address,String Age){
        this.Name = Name;
        this.Gender = Gender;
        this.Address = Address;
        this.Age = Age;
    }

}
``` |
| Main.java | ```
public class Main {

    public static void main(String[] args)
    {
        int count;
        Scanner scanner = new Scanner(System.in);
        System.out.print("Input the number of teachers required : ");
        count = scanner.nextInt();
        Teacher[] teachers = new Teacher[count];
``` |

```
        for(int i=0;i<count;i++) {
          int j = i+1;
          System.out.print("Enter Empid of teacher "+j+" : ");
          String Empid = scanner.next();
          System.out.print("Enter Name of teacher "+j+" : ");
          String Name = scanner.next();
          System.out.print("Enter Gender of teacher "+j+" : ");
          String Gender = scanner.next();
          System.out.print("Enter Age of teacher "+j+" : ");
          String Age = scanner.next();
          System.out.print("Enter Salary of teacher "+j+" : ");
          String Salary = scanner.next();
          System.out.print("Enter Address of teacher "+j+" : ");
          String Address = scanner.next();
          System.out.print("Enter department of teacher "+j+" : ");
          String department = scanner.next();
          System.out.print("Enter Subjects of teacher "+j+" : ");
          String Subjects = scanner.next();

          teachers[i] = new
Teacher(Name,Gender,Address,Age,Empid,Salary,department,Subjects);
        }

        System.out.println("\n-------------------------------------------------------
----------------\n");
        System.out.println("Teacher's List \n");


        for(int i=0;i<count;i++) {
          teachers[i].displayTeacherDetails();
        }

    }

}
```

RESULT: The above program is executed and obtained the output.

OUTPUT:

```
Run:    Main (1) ×

    Input the number of teachers required : 2
    Enter Empid of teacher 1 : 100
    Enter Name of teacher 1 : Jacob
    Enter Gender of teacher 1 : male
    Enter Age of teacher 1 : 28
    Enter Salary of teacher 1 : 20000
    Enter Address of teacher 1 : Delhi
    Enter department of teacher 1 : Maths
    Enter Subjects of teacher 1 : Geometry
    Enter Empid of teacher 2 : 101
    Enter Name of teacher 2 : Gemini
    Enter Gender of teacher 2 : Female
    Enter Age of teacher 2 : 31
    Enter Salary of teacher 2 : 30000
    Enter Address of teacher 2 : Mumbai
    Enter department of teacher 2 : Maths
    Enter Subjects of teacher 2 : Algebra


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

    Teacher's List

    Empid : 100 Name : Jacob Gender : male Age : 28 Salary : 20000 Address : Delhidepartment : Maths Subjects : Geometry
    Empid : 101 Name : Gemini Gender : Female Age : 31 Salary : 30000 Address : Mumbaidepartment : Maths Subjects : Algebra
```

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 named 'Publisher' with data members pname, pid; a constructor named

Publisher().

Step 3: Create a class named 'Book' which is derived 'Publisher' with data members nop, price; a constructor named Book().

Step 4: Create a class named 'literature' which is derived from Book with data members title, author; a constructor; a function show() to display details. Step 5: Create a class named 'fiction' which is derived from Book with data members bname, auth; a constructor; a function display() to print details. Step 6: Print a menu defining the type of genres; if literatute create an object of literature

type and object of type fiction if fiction is chosen.

Step 5 : Stop

PROGRAM CODE:

Book.java	public class Book extends Publisher{
	String book = "Mysterious Woods";
	}
Fiction.java	public class Fiction extends Book{
	String genre = "Fiction";
	}
Literature.java	public class Literature extends Book{
	String genre = "Literature";
	}
Publisher.java	public class Publisher {

35

	String publisher = "Wiley"; }

Main.java	```java
	public class Main {
	
	    public static void main(String[] args) {
	
	        Literature literatureBook = new Literature();
	        Fiction fictionBook = new Fiction();
	
	        System.out.println("Fiction Book Details : ");
	        System.out.println("Fiction Book Publisher :
	"+fictionBook.publisher);
	        System.out.println("Fiction Book Name : "+fictionBook.book);
	        System.out.println("Fiction Book Genre : "+fictionBook.genre);
	
	    }
	
	}```

RESULT: The above program is executed and obtained the output.

OUTPUT:

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 a class named 'Student' with data members as subject names; a constructor.

Step 3: Create a class named 'Sports' which is derived 'Student' with data members

goals,matches,won; a constructor.

Step 4: Create a class named 'Result' which is derived from Sports member functions

displayStud() and displaySport() to display details.

Step 5: Create an 3 objects of type Student, Sports and Result, and display the details.

Step 6: Stop

PROGRAM CODE:

Student.java	`public class Student {`
	
	`    int maths = 85;`
	`    int science = 72;`
	`    int english = 88;`
	`    int socialScience = 70;`
	
	`}`

Sports.java	`public class Sports extends Student {`
	
	`    String sport = "Football";`
	`    int goals = 2;`
	`    int assists = 1;`
	`    int minutesPlayed = 81;`
	`    int grace = 20;`
	
	`}`

Result.java	`public class Result extends Sports {`
	
	`    public void displayInfo()`
	`    {`
	`        System.out.println("Academic Result");`

| | ```java
System.out.println("---------------");
System.out.println("Maths : "+this.maths);
System.out.println("Science : "+this.science);
System.out.println("English : "+this.english);
System.out.println("Social Science : "+this.socialScience);
System.out.println("Sports Grace : "+this.grace);
System.out.println("---------------");


System.out.println("\n");
System.out.println("Sports Result");
System.out.println("-------------");
System.out.println("Sport : "+this.sport);
System.out.println("Goals : "+this.goals);
System.out.println("Assists : "+this.assists);
System.out.println("Minutes Played "+this.minutesPlayed);
System.out.println("-------------");
    }
}
``` |
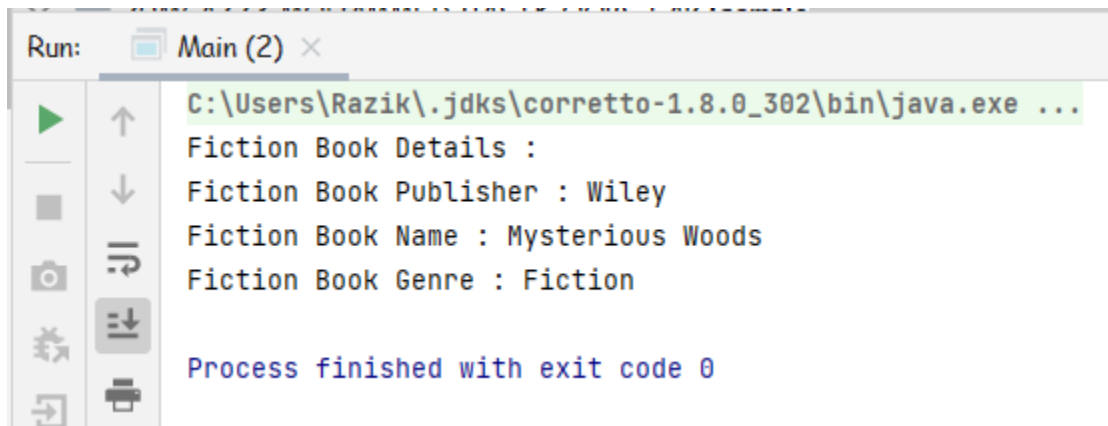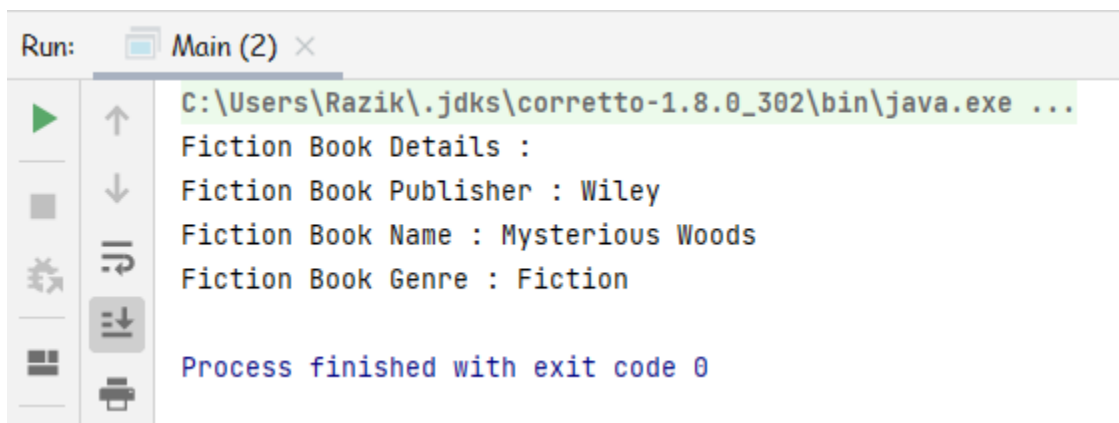| Main.java | ```java
public class Main {

    public static void main(String[] args) {

        Result result = new Result();
        result.displayInfo();

    }
}
``` |

RESULT: The above program is executed and obtained the output.

OUTPUT:

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 an interface 'find' with 2 functions area() and perimeter().

Step 3: Create a class named circle to implement the functions of inteface to find area and

perimeter of circle.

Step 4: Create a class named rect to implement the functions of inteface to find area and

perimeter of rectangle.

Step 5: Create objects for both these classes and call functions area() and perimeter() to

display the same.

Step 6:Stop.

PROGRAM CODE:

Circle.java	public class Circle implements Shape{

	int radius;
	Scanner scanner = new Scanner(System.in);
	
	public void perimeter() {
	   System.out.println("Input radius of circle : ");
	   radius = scanner.nextInt();
	   String perimeter = Double.toString(Math.PI*radius*2);
	   System.out.println("Circumference of the circle is : "+perimeter);
	}
	
	public void area() {
	   System.out.println("Input radius of circle : ");
	   radius = scanner.nextInt();
	   String area = Double.toString(Math.PI*radius*radius);
	   System.out.println("Area of the circle is : "+area);
	}

	}
| Rectangle.java | ```java
import java.util.Scanner;
public class Rectangle implements Shape{

    int length;
    int breadth;

    Scanner scanner = new Scanner(System.in);

    public void perimeter() {
        System.out.println("Input length of rectangle : ");
        length = scanner.nextInt();
        System.out.println("Input breadth of rectangle : ");
        length = scanner.nextInt();
        String perimeter = Double.toString(2*(length+breadth));
        System.out.println("Perimeter of the rectangle is : "+perimeter);
    }

    public void area() {
        System.out.println("Input length of rectangle : ");
        length = scanner.nextInt();
        System.out.println("Input breadth of rectangle : ");
        length = scanner.nextInt();
        String area = Double.toString(length*breadth);
        System.out.println("Area of the rectangle is : "+area);
    }

}
``` |
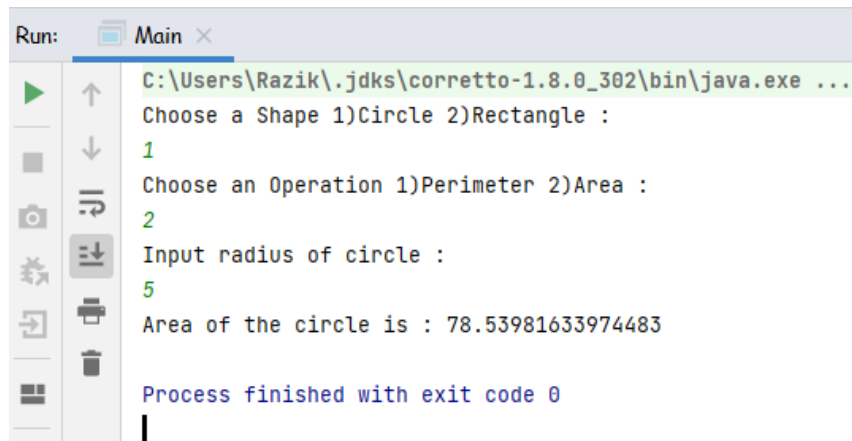| Shape.java | ```java
public interface Shape {
    public void perimeter();
    public void area();
}
``` |
| Main.java | ```java
import java.util.Scanner;

public class Main {

    public static void main(String[] args){

        Scanner scanner = new Scanner(System.in);
        int shape,operation;

        System.out.println("Choose a Shape 1)Circle 2)Rectangle : ");
        shape = scanner.nextInt();
``` |

40

```java
            System.out.println("Choose an Operation 1)Perimeter 2)Area : ");
            operation = scanner.nextInt();

            if(shape==1){
                Circle circle = new Circle();
                if(operation==1){
                    circle.perimeter();
                }
                else if(operation==2)
                {
                    circle.area();
                }
                else {
                    System.out.println("Operation code.");
                }
            }
            else if(shape==2)
            {
                Rectangle rectangle = new Rectangle();
                if(operation==1){
                    rectangle.perimeter();
                }
                else if(operation==2)
                {
                    rectangle.area();
                }
                else {
                    System.out.println("Operation code :");
                    System.exit(0);
                }
            }
            else {
                System.out.println("Incorrect Shape code.");
            }

        }

}
```

RESULT: The above program is executed and obtained the output.

OUTPUT:

```
Run:    Main ×
    C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
    Choose a Shape 1)Circle 2)Rectangle :
    1
    Choose an Operation 1)Perimeter 2)Area :
    2
    Input radius of circle :
    5
    Area of the circle is : 78.53981633974483

    Process finished with exit code 0
```

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: Create an interface 'calculatedemo' with function calculate().

Step 3: Create a class named 'billcalc' that implements the function to print the bill in required manner.

Step 4: Create an object of type billcalc to print the bill.

Step 5: Stop.

PROGRAM CODE:

Bill.java	public interface Bill {

	String productId = "";
	String productName="";
	int unitPrice = 0;
	int quantity = 0;

	```java int total = 0; public void printBillItem(); public void printBillHeader(); public void printBillFooter(int billTotal); } ```
ProductBill.java	```java import java.time.format.DateTimeFormatter; import java.time.LocalDateTime; public class ProductBill implements Bill { String productId = ""; String productName=""; int unitPrice = 0; int quantity = 0; int total = 0; ProductBill(String productId,String productName,int unitPrice,int quantity){ this.productId = productId; this.productName = productName; this.unitPrice = unitPrice; this.quantity = quantity; this.total = unitPrice*quantity; } public void printBillHeader() { System.out.println("Order No : " + Math.random() * 1000); DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy/MM/dd HH:mm:ss"); LocalDateTime now = LocalDateTime.now(); System.out.println("Date : " + dtf.format(now)); System.out.println("Product ID Name Quantity Unit Price Total "); System.out.println("--"); } public void printBillItem() { System.out.format("%10s%20s%10d%12d%12d \n",this.productId,this.productName,this.unitPrice,this.quantity,this.total); } ```

44

| | ```java
public void printBillFooter(int billTotal)
{
  System.out.println("----------------------------------------------------------------");
  System.out.format("%64s \n","Net. Amount : "+billTotal);
}

}
``` |
| Main.java | ```java
import java.util.Scanner;

public class Main {

  public static void main(String[] args)
  {

    Scanner scanner = new Scanner(System.in);

    System.out.println("Input Number of items in Bill : ");
    int count = scanner.nextInt();
    ProductBill[] productBill=new ProductBill[count];
    int billTotal=0;

    for(int i=0;i<count;i++) {
      System.out.println("Enter Product ID : ");
      String productId = scanner.next();
      System.out.println("Enter Name : ");
      String name = scanner.next();
      System.out.println("Enter Quantity : ");
      int qty = scanner.nextInt();
      System.out.println("Unit Price : ");
      int up = scanner.nextInt();

      productBill[i]=new ProductBill(productId,name,up,qty);
    }

    if(count>0){

      productBill[0].printBillHeader();

      for(int i=0;i<count;i++) {
        productBill[i].printBillItem();
        billTotal += productBill[i].total;
      }
      productBill[0].printBillFooter(billTotal);
``` |

	} } }

RESULT: The above program is executed and obtained the output.

OUTPUT:

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
Input Number of items in Bill :
3
Enter Product ID :
101
Enter Name :
Jam
Enter Quantity :
2
Unit Price :
30
Enter Product ID :
231
Enter Name :
Bread
Enter Quantity :
1
Unit Price :
25
Enter Product ID :
561
Enter Name :
Butter
Enter Product ID :
561
Enter Name :
Butter
Enter Quantity :
1
Unit Price :
40
Order No : 592.8991713463721
Date : 2021/09/09 12:41:56
Product ID          Name  Quantity  Unit Price      Total
------------------------------------------------------------
     101             Jam      30          2           60
     231           Bread      25          1           25
     561          Butter      40          1           40
------------------------------------------------------------
                                        Net. Amount : 125

Process finished with exit code 0
```

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: To create a package named graphics, create a folder of the same name in the

directory. Here inside that we have another module named calculate

Step 3: Inside the graphics folder, create modules for finding the areas of rectangle, circle, triangle and square.

Step 4: Outside the graphics folder, write a program to access the modules mention above

and print the output.

Step 5: Stop

PROGRAM CODE:

| Graphics/ Circle.java | ```java
import java.util.Scanner;

public class Circle implements Shape {

   int radius;
   Scanner scanner = new Scanner(System.in);

   public void area() {
      System.out.println("Input radius of circle : ");
      radius = scanner.nextInt();
      String area = Double.toString(Math.PI*radius*radius);
      System.out.println("Area of the circle is : "+area);
   }

}
``` |
|---|---|
| Graphics/ Rectangle.java | ```java
import java.util.Scanner;

public class Rectangle implements Shape {

   int length;
``` |

| | ```java
int breadth;

Scanner scanner = new Scanner(System.in);

public void area() {
   System.out.println("Input length of rectangle : ");
   length = scanner.nextInt();
   System.out.println("Input breadth of rectangle : ");
   length = scanner.nextInt();
   String area = Double.toString(length*breadth);
   System.out.println("Area of the rectangle is : "+area);
}

}
``` |
| Graphics/ Shape.java | ```java
public interface Shape {
   public void area();
}
``` |
| Graphics/ Square.java | ```java
import java.util.Scanner;

public class Square {

   int side;
   Scanner scanner = new Scanner(System.in);

   public void area() {
      System.out.println("Input side length of square : ");
      side = scanner.nextInt();
      String area = Double.toString(side*side);
      System.out.println("Area of the square : "+area);
   }

}
``` |
| Graphics/ Triangle.java | ```java
import java.util.Scanner;

public class Triangle {

   int height;
   int breadth;

   Scanner scanner = new Scanner(System.in);

   public void area() {

      System.out.println("Input height of the triangle : ");
      height = scanner.nextInt();
``` |

| | ```java
System.out.println("Input breadth of triangle : ");
breadth = scanner.nextInt();
String area = Double.toString((height*breadth)/2f);
System.out.println("Area of the triangle is : "+area);

    }

}
``` |
| Main.java | ```java
import com.lab_cycles.co4.q1.Graphics.Circle;
import com.lab_cycles.co4.q1.Graphics.Rectangle;
import com.lab_cycles.co4.q1.Graphics.Square;
import com.lab_cycles.co4.q1.Graphics.Triangle;

import java.util.Scanner;

public class Main {

    public static void main(String[] args){
        Scanner scanner = new Scanner(System.in);
        int shape;

        System.out.println("Choose a Shape 1)Circle 2)Rectangle 3)Square
4)Triangle : ");
        shape = scanner.nextInt();

        if(shape==1){
            Circle circle = new Circle();
            circle.area();
        }
        else if(shape==2)
        {
            Rectangle rectangle = new Rectangle();
            rectangle.area();
        }
        else if(shape==3)
        {
            Square square = new Square();
            square.area();
        }
        else if(shape==4)
        {
            Triangle triangle = new Triangle();
            triangle.area();
        }
        else {
            System.out.println("Incorrect Shape code.");
``` |

	}
	}
	}

RESULT: The above program is executed and obtained the output.

OUTPUT:

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
Connected to the target VM, address: '127.0.0.1:61561', transport: 'socket'
Choose a Shape 1)Circle 2)Rectangle 3)Square 4)Triangle :
1
Input radius of circle :
10
Area of the circle is : 314.1592653589793
Disconnected from the target VM, address: '127.0.0.1:61561', transport: 'socket'

Process finished with exit code 0
```

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
Choose a Shape 1)Circle 2)Rectangle 3)Square 4)Triangle :
4
Input height of the triangle :
40
Input breadth of triangle :
23
Area of the triangle is : 460.0

Process finished with exit code 0
```

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: To create a package named arithmetic, create a folder of the same name in the

directory. Here inside that we have another module named operation

Step 3: Inside arithmetic package, create modules to perform addition, subtraction, multiplication and division of 2 numbers.

Step 4: Outside the folder, write another program that acssess the above module and print

the output.

Step 5:Stop

PROGRAM CODE:

| Arithmetic/ AdditionOper ation.java | ```
public class AdditionOperation implements ArithmeticOperation {

    public int operateNumbers(int number1,int number2)
    {
       return number1+number2;
    }

}
``` |
|---|---|
| Arithmetic/ ArithmeticO peration.java | ```
public interface ArithmeticOperation {

    public int operateNumbers(int number1,int number2);

}
``` |
| Arithmetic/ DivisionOper ator.java | ```
public class DivisionOperator implements ArithmeticOperation{

    public int operateNumbers(int number1,int number2)
    {
``` |

	``
	      return number1*number2;
	   }
	
	}
	``

Arithmetic/ MultiplicationOperator.java	```java
	public class MultiplicationOperator implements ArithmeticOperation{
	
	   public int operateNumbers(int number1,int number2)
	   {
	      return number1*number2;
	   }
	
	}
	``

Arithmetic/ SubtractionOperation.java	```java
	public class SubtractionOperation implements ArithmeticOperation{
	
	   public int operateNumbers(int number1,int number2)
	   {
	      return number1-number2;
	   }
	
	}
	``

Main.java	```java
	import com.lab_cycles.co4.q2.Arithmetic.AdditionOperation;
	import com.lab_cycles.co4.q2.Arithmetic.DivisionOperator;
	import com.lab_cycles.co4.q2.Arithmetic.MultiplicationOperator;
	import com.lab_cycles.co4.q2.Arithmetic.SubtractionOperation;
	
	import java.util.Scanner;
	
	public class Main {
	
	   public static void main(String[] args)
	   {
	
	      Scanner scanner = new Scanner(System.in);
	
	      System.out.println("Choose an operation : 1.Add, 2.Subtract, 3.Multiply, 4.Divide");
	      int choice = scanner.nextInt();
	
	      System.out.println("Enter the numbers : ");
	      int number1 = scanner.nextInt();
	      int number2= scanner.nextInt();
	
	      System.out.println("The Result is : 3");
	
	      switch (choice){
	``

<table>
<tr><td></td><td>

```
            case 1:
                AdditionOperation additionOperation = new
AdditionOperation();

System.out.println(additionOperation.operateNumbers(number1,number2)
);
                break;
            case 2:
                SubtractionOperation subtractionOperation = new
SubtractionOperation();

System.out.println(subtractionOperation.operateNumbers(number1,numbe
r2));
                break;
            case 3:
                MultiplicationOperator multiplicationOperator = new
MultiplicationOperator();

System.out.println(multiplicationOperator.operateNumbers(number1,numb
er2));
                break;
            case 4:
                DivisionOperator divisionOperator = new DivisionOperator();

System.out.println(divisionOperator.operateNumbers(number1,number2));
                break;
            default:
                System.out.println("Invalid Code");
        }

    }

}
```

</td></tr>
</table>

RESULT: The above program is executed and obtained the output.

OUTPUT:

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
Choose an operation : 1.Add, 2.Subtract, 3.Multiply, 4.Divide
1
Enter the numbers :
3
78
The Result is : 3
81

Process finished with exit code 0
```

PROGRAM NO : 19

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

ALGORITHM :

Step 1: Start

Step 2: Create a class named usernameex that inherits Exception class with a constructor that

calls Exception class constructor and pass error meaasage.

Step 3: Create a class named passwordex that inherits Exception class with a constructor that

calls Exception class constructor and pass error meaasage.

Step 4: Inside the main(), Read the username and password.

Step 5: Inside the try block, we throw usernamex and passwordex with appropriate message

if any of the conditon is true:

- If username is empty
- If password is empty
- If password doesnt contain special charecters
- If username length is less than 6
- If password is not string enough

Step 6: Inside the catch block with parameter usernameex's object, print "USERNAME

EXCEPTION OCCURED"

Step 7: Inside the catch block with parameter passwordex's object, print "PASSWORD

EXCEPTION OCCURED"

Step 8:Stop

PROGRAM CODE:

UserExceptio n.java	public class UserException extends Exception {
	
	   public UserException(String s)
	   {
	     // Call constructor of parent Exception
	     super(s);
	   }

	}
| Main.java | ```java
import java.util.Scanner;

public class Main
{
    public static void main(String args[])
    {

        Scanner scanner = new Scanner(System.in);
        String username,password;

        try
        {
            System.out.println("Please enter a username : ");
            username = scanner.nextLine();

            if(username.equals("")){
                throw new UserException("Username not provided !");
            }

            if(username.length()<8){
                throw new UserException("The username should be
atleast 8 characters long !");
            }


            System.out.println("Please enter a password : ");
            password = scanner.nextLine();

            if(password.equals("")){
                throw new UserException("Password not provided !");
            }

            if(password.length()<6){
                throw new UserException("The password should be
atleast 6 characters long !");
            }

            System.out.println("User Data Accepted");

        }
        catch (UserException ex)
        {
            System.out.println("Caught");
``` |

	System.out.println(ex.getMessage());
	        }
	      }
	    }

RESULT: The above program is executed and obtained the output.

OUTPUT:

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
Please enter a username :
jami
Caught
The username should be atleast 8 characters long !

Process finished with exit code 0
```

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: Create a class named NegException that inherits Exception class with a constructor

inside which we call the Exception class constructor and pass error meaasage.

Step 3: Inside the main(), Read the limit of array

Step 4: Inside the try block,read the array and check if any element is less than 0

Step 5: If true, throw NegException with appropriate message.

Step 6: Calculate the average of the array and print it

Step 7: Inside the catch exception, Print "NEGETIVE EXCEPTION OCCURED"

Step 8:Stop.

PROGRAM CODE:

| NegativeNumberException.java | ```java
public class NegativeNumberException extends Exception {

    public NegativeNumberException(String s)
    {
      // Call constructor of parent Exception
      super(s);
    }

}
``` |
|---|---|
| Main.java | ```java
import java.util.Scanner;

public class Main {
  public static void main(String args[]) {

      Scanner scanner = new Scanner(System.in);
      int count;
``` |

```
        System.out.println("Please enter the number of numbers to be
calculated : ");
        count = scanner.nextInt();

        int[] numbers = new int[count];

        try {
          System.out.println("Please enter " + count + " numbers : ");

          for (int i = 0; i < count; i++) {
            int num = scanner.nextInt();
            numbers[i] = num;

            if (num < 0) {
                throw new NegativeNumberException("Negative numbers are
not allowed.");
            }

          }

          int sum =0;

          for (int i = 0; i < count; i++) {
            sum += numbers[i];
          }

          float average = (float)sum/count;
          System.out.println("Average of given numbers is : "+average);



        } catch (NegativeNumberException ex) {
          System.out.println("Exception Caught !!");
          System.out.println(ex.getMessage());
        }

    }
}
```

RESULT: The above program is executed and obtained the output.

OUTPUT:

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
Please enter the number of numbers to be calculated :
4
Please enter 4 numbers :
34
65
21
-90
Exception Caught !!
Negative numbers are not allowed.

Process finished with exit code 0
```

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: Create a class named mul that inherits Thread class with member function as run()

Step 3: Inside run(), Print the multiplication table for 5

Step 4: Create a class named prime that inherits Thread class with memebr function run()

Step 5: Inside run(),Print the prime numbers upto the limit of user's choice

Step 6: Inised the main(), create an object for the classes and call start() using each object

Step 7: Stop.

PROGRAM CODE:

| PrimeNumbers.java | ```java
import java.util.Scanner;

public class PrimeNumbers extends Thread {

    int count;
    int primeCount = 0;

    PrimeNumbers(int count){
        this.count=count;
    }

    public void run() {

        for(int i=1;primeCount<count;i++){
            boolean isPrime=true;
            if(i==1)
            {
                i++;
            }
``` |
|---|---|

	```for (int j=2;j<i;j++){```
	``` if(i%j==0){```
	``` isPrime = false;```
	``` break;```
	``` }```
	```}```
	
	```if (isPrime){```
	``` System.out.println("Prime Number : "+i);```
	``` primeCount++;```
	```}```
	``` }```
	
	``` }```
	```}```

MultiTable5.java	```public class MultiTable5 extends Thread {```
	
	``` int count;```
	
	``` MultiTable5(int count){```
	``` this.count=count;```
	``` }```
	
	``` public void run() {```
	``` for (int i=0;i<=count;i++){```
	``` System.out.println("5 x "+i+" = "+(5*i));```
	``` }```
	
	``` }```
	
	```}```

Main.java	```import java.util.Scanner;```
	
	```public class Main {```
	
	``` public static void main(String[] args)```
	``` {```
	
	``` Scanner scanner = new Scanner(System.in);```
	``` int primeCount,multipleCount;```
	
	``` System.out.println("Enter number of Prime Numbers to be listed : ");```

```
                    primeCount = scanner.nextInt();
                    System.out.println("Enter number of Multiples of 5 to be listed : ");
                    multipleCount = scanner.nextInt();

                    PrimeNumbers thread1 = new PrimeNumbers(primeCount);
                    thread1.start();

                    MultiTable5 thread2 = new MultiTable5(multipleCount);
                    thread2.start();

                 }

           }
```

RESULT: The above program is executed and obtained the output.

OUTPUT:

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
Enter number of Prime Numbers to be listed :
3
Enter number of Multiples of 5 to be listed :
10
Prime Number : 2
Prime Number : 3
Prime Number : 5
5 x 0 = 0
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50

Process finished with exit code 0
```

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: Create a class named even that implements Runnable interface with function run()

Step 3: Inside run(), we read the limit for printing even numbers and print it using for loop. Step 4:Create another calss fib that implements Runnable interface with function run().

Step 5: Inside run(), Initialise n1 as 0,n2 as 1 and n3 as 0.

Step 6: Check if n<0, if true, print "Enter a positive number" else go to step 7

Step 7: Repeat step8 to 11 until n3>n

Step 8: Print n1

Step 9: n3=n1+n2

Step 10:n1=n2

Step 11: n2=n3

Step 12: Create object e of even and create an object t1 of Thread with its parameterized

constructor passing e as parameter

Step 13: Call start() using t1

Step 14: Do the same for class odd with Thread object t2 and call start() using t2

Step 15: Stop

PROGRAM CODE:

EvenSeries.java	public class EvenSeries implements Runnable {
	
	    int count;
	    int[] fibonacciSeries;

63

| | ```
    EvenSeries(int count){this.count=count;}

    public void run() {

       for (int i=0;i<count;i++)
       {
         if(i%2==0)
         {
            System.out.println("Even Number : "+i);
         }
       }

    }

}
``` |
| FibonacciSeri es.java | ```
public class FibonacciSeries implements Runnable{

   int count;
   long[] fibonacciSeries;

   FibonacciSeries(int count){this.count=count;this.fibonacciSeries=new
long[count];}

   public void run() {

      this.fibonacciSeries[0]=0;
      this.fibonacciSeries[1]=1;
      this.fibonacciSeries[2]=1;

      for (int i=3;i<count;i++)
      {
         this.fibonacciSeries[i]= this.fibonacciSeries[(i-1)]+
this.fibonacciSeries[(i-2)];

         if(i>91) {
            break;
         }

         System.out.println("Fibonacci Number : " +
this.fibonacciSeries[i]);
      }

   }
``` |

	}
| Main.java | ```java
import java.util.Scanner;

public class Main {

    public static void main(String[] args)
    {

        Scanner scanner = new Scanner(System.in);
        int fiboCount,evenCount;

        System.out.println("Enter number of Fibonacci Numbers to be listed : ");
        fiboCount = scanner.nextInt();
        System.out.println("Enter number of Even Numbers to be listed : ");
        evenCount = scanner.nextInt();

        FibonacciSeries fibonacciSeries = new FibonacciSeries(fiboCount);
        EvenSeries evenSeries = new EvenSeries(evenCount);

        Thread thread1 = new Thread(evenSeries);
        Thread thread2 = new Thread(fibonacciSeries);

        thread1.start();
        thread2.start();

    }

}
``` |

RESULT: The above program is executed and obtained the output.

OUTPUT:

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
Enter number of Fibonacci Numbers to be listed :
120
Enter number of Even Numbers to be listed :
100
Even Number : 0
Even Number : 2
Even Number : 4
Even Number : 6
Even Number : 8
Even Number : 10
Fibonacci Number : 2
Fibonacci Number : 3
Fibonacci Number : 5
Fibonacci Number : 8
Fibonacci Number : 13
```

PROGRAM NO : 23

AIM: Producer/Consumer using ITC

ALGORITHM :

Step 1:Start

Step 2:In PC class (A class that has both produce and consume methods), a linked list of

jobs and a capacity of the list is added to check that producer does not produce if the list is

full.

Step 3:In Producer class, the value is initialized as 0.

Step 4:We have an infinite outer loop to insert values in the list. Inside this loop, we have a

synchronized block so that only a producer or a consumer thread runs at a time. An inner loop is there before adding the jobs to list that checks if the job list is full, the

producer thread gives up the intrinsic lock on PC and goes on the waiting state.

Step 5:If the list is empty, the control passes to below the loop and it adds a value in the list.

Step 6:In the Consumer class, we again have an infinite loop to extract a value from the list. Inside, we also have an inner loop which checks if the list is empty.

Step 7:If it is empty then we make the consumer thread give up the lock on PC and passes

the control to producer thread for producing more jobs.

Step 8:If the list is not empty, we go round the loop and removes an item from the list.

Step 9:In both the methods, we use notify at the end of all statements. The reason is simple, once you have something in list, you can have the consumer thread consume it, or if you

have consumed something, you can have the producer produce something.

Step 10:sleep() at the end of both methods just make the output of program run in step wise

manner and not display everything all at once so that you can see what actually is happening

in the program.

Step 11:Stop.

PROGRAM CODE:

Main.java	
| | ```java
import java.util.LinkedList;

public class Main {

  public static void main(String[] args) throws InterruptedException {

    final PC pc = new PC();


    Thread t1 = new Thread(new Runnable() {
      @Override
      public void run() {
        try {
          pc.produce();
        } catch (InterruptedException e) {
          e.printStackTrace();
        }
      }
    });


    Thread t2 = new Thread(new Runnable() {
      @Override
      public void run() {
        try {
          pc.consume();
        } catch (InterruptedException e) {
          e.printStackTrace();
        }
      }
    });


    t1.start();
    t2.start();


    t1.join();
    t2.join();
  }
``` |

```java
public static class PC {

    LinkedList<Integer> list = new LinkedList<>();
    int capacity = 2;

    public void produce() throws InterruptedException {
        int value = 0;
        while (true) {
            synchronized (this) {

                while (list.size() == capacity)
                    wait();

                System.out.println("Producer produced-"
                        + value);

                list.add(value++);

                notify();

                Thread.sleep(1000);
            }
        }
    }

    public void consume() throws InterruptedException {
        while (true) {
            synchronized (this) {

                while (list.size() == 0)
                    wait();

                int val = list.removeFirst();

                System.out.println("Consumer consumed-"
                        + val);

                notify();
```

```
                    Thread.sleep(1000);
                 }
              }
          }
      }
  }      this.count=count;
        }

     public void run() {

        for(int i=1;primeCount<count;i++){
           boolean isPrime=true;
           if(i==1)
           {
              i++;
           }

           for (int j=2;j<i;j++){
              if(i%j==0){
                 isPrime = false;
                 break;
              }
           }
           if (isPrime){
              System.out.println("Prime Number : "+i);
              primeCount++;
           }
        }
     }
  }
```

RESULT: The above program is executed and obtained the output.

OUTPUT :

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
Producer produced-0
Consumer consumed-0
Producer produced-1
Producer produced-2
Consumer consumed-1
Consumer consumed-2
Producer produced-3
```

PROGRAM NO : 24

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

ALGORITHM :

Step 1: Start

Step 2: Create a class named stack with data members as a(an array),top(set as -1),ch,item,i;

a function named menu()

Step 3: Inside menu(), give choices to push,pop and display the stack

Step 4: If the choice is 1, then check whether the stack is full, else add an element into the

stack.

Step 5: If the choice is 2, then check whether the stack is empty, else delete an element into

the stack.

Step 6: If the choice is 3, then check whether the stack is empty, else print all the elements in

the stack.

Step 7: If the choice is greater than 4, then print "Invalid option".

Step 8: Inside the main(), create an object of type stack and call the menu() function.

Step 9:Stop.

PROGRAM CODE:

Main.java	
	``import java.util.Scanner;

class Main{

public static void main(String[] args) {

   Scanner scanner = new Scanner(System.in);
   System.out.println("Enter size of stack : ");
   int count = scanner.nextInt();
   GenericStack stack = new GenericStack(count);

   while (true) {
      System.out.println("Enter operation : 1)Push 2)Pop 3)Display : ");
      int choice = scanner.nextInt();
      switch (choice) {
         case 1:``` |

| | ```
          System.out.println("item to insert :");
          int item = scanner.nextInt();
          stack.push(item);
          break;
        case 2:
          stack.pop();
          break;
        case 3:
          stack.printStack();
      }
    }
  }
}
``` |
| GenericStack<br>.java | ```
class GenericStack {
  private int[] arr;
  private int top;
  private int capacity;

  // Creating a stack
  GenericStack(int size) {
    arr = new int[size];
    capacity = size;
    top = -1;
  }

  // Add elements into stack
  public void push(int x) {
    if (isFull()) {
      System.out.println("OverFlow");
    }
    else{
      System.out.println("Inserting " + x);
      arr[++top] = x;
    }

  }

  // Remove element from stack
  public int pop() {
    if (isEmpty()) {
      System.out.println("STACK EMPTY");
      return -1;
    }
    else {
``` |

71

```
                return arr[top--];
            }
        }
        // Utility function to return the size of the stack
        public int size() {
            return top + 1;
        }
        // Check if the stack is empty
        public Boolean isEmpty() {
            return top == -1;
        }
        // Check if the stack is full
        public Boolean isFull() {
            return top == capacity - 1;
        }
        public void printStack() {
            for (int i = 0; i <= top; i++) {
                System.out.println(arr[i]);
            }
        }
    }
```

RESULT: The above program is executed and obtained the output.

OUTPUT :

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
Enter size of stack :
4
Enter operation : 1)Push 2)Pop 3)Display :
1
item to insert :
34
Inserting 34
Enter operation : 1)Push 2)Pop 3)Display :
1
item to insert :
67
Inserting 67
Enter operation : 1)Push 2)Pop 3)Display :
3
34
67
Enter operation : 1)Push 2)Pop 3)Display :
2
Enter operation : 1)Push 2)Pop 3)Display :
3
34
Enter operation : 1)Push 2)Pop 3)Display :
```

PROGRAM NO : 25

AIM: Using generic method perform Bubble sort.

ALGORITHM :

Step 1: Start

Step 2: Create a function named bubblesort(array)

Step 3: n<- length of array

Step 4: Intialize temp<-0

Step 5: i<-0

Step 6:Reapeat steps from to until i>n

Step 7: j<-1,repeat the steps from to until j>n-I

Step 8: check if array[i] >array[j], if true,swap them;else increment j

Step 9: Inside main () Initialize an array with elements and the print the same

Step 10: Call the function bubblesort() and pass the array as parameter

Step 11: Print the sorted array

Step 12: Stop.

PROGRAM CODE:

Main.java	
| | ```
import java.util.Arrays;
import java.util.Scanner;


public class Main {
    static void bubbleSort(int array[]) {
        int size = array.length;
        for (int i = 0; i < size - 1; i++)
            for (int j = 0; j < size - i - 1; j++)
                if (array[j] > array[j + 1]) {

                    int temp = array[j];
                    array[j] = array[j + 1];
                    array[j + 1] = temp;
``` |

```
                }
            }

        public static void main(String args[]) {

            Scanner scanner = new Scanner(System.in);

            System.out.println("Number of items to be inserted : ");
            int count = scanner.nextInt();

            int[] data = new int[count];

            System.out.println("Enter the array items : ");

            for(int i=0;i<count;i++)
            {
                data[i] = scanner.nextInt();
            }

            Main.bubbleSort(data);

            System.out.println("Sorted Array in Ascending Order:");
            System.out.println(Arrays.toString(data));
        }
    }
```

RESULT: The above program is executed and obtained the output.

OUTPUT :

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
Number of items to be inserted :
6
Enter the array items :
78
45
56
21
10
71
Sorted Array in Ascending Order:
[10, 21, 45, 56, 71, 78]

Process finished with exit code 0
```

PROGRAM NO : 26

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

ALGORITHM :

Step 1: Start

Step 2: Create an object 'a' of type ArrayList. Step 3: Put values into it using add()

Step 4: Manipulate the list using built in functions. Step 5: Print the elements in a

Step 6: Stop.

PROGRAM CODE:

Main.java	
| | ```java
import java.util.*;
public class Main {
    public static void main(String[] args) {

        // Creating ArrayList of type "String" which means we can only add "String" elements

        ArrayList<String> fruits = new ArrayList<String>();

        //adding elements to an ArrayList

        fruits.add("Pomegranate");
        fruits.add("Lemon");
        fruits.add("Avocado");
        fruits.add("Durian");
        fruits.add("Watermelon");
        fruits.add(3, "Orange");


        // Displaying elements

        System.out.println("\n ORIGINAL LIST:");
        System.out.println("-----------------------------------------------------");
        for(String str : fruits)
            System.out.printf(str+"  ");
``` |

```
//Remove elements from ArrayList

fruits.remove("Avocado");
fruits.remove(2);

// Displaying elements
System.out.println("\n------------------------------------------------------");
System.out.println("\n\nARRAYLIST AFTER REMOVAL OF
ELEMENTS:");
System.out.println("------------------------------------");
for(String str : fruits )
   System.out.printf(str+"  ");


//Updating the ArrayList

fruits.set(3,"Guava");
System.out.println("\n------------------------------------");
System.out.println("\n\n ARRAYLIST AFTER UPDATION:");
System.out.println("------------------------------------");
for(String str : fruits )
   System.out.printf(str+"  ");
System.out.println("\n------------------------------------");
//Sorting the ArrayList

Collections.sort(fruits);

System.out.println("\n\n ARRAYLIST AFTER SORTING:");
System.out.println("------------------------------------");
for (String str : fruits)
   System.out.printf(str+"  ");



// Checks whether the object is in the ArrayList
System.out.println("\n------------------------------------------");
System.out.println("\nApple is in the List- "+
fruits.contains("Durian"));
System.out.println("Strawberry is in the List-
"+fruits.contains("Strawberry"));


//Size of the ArrayList

System.out.println("\n------------------------------------------");
System.out.println("\nSIZE OF THE ARRAYLIST: "+ fruits.size());
```

```
                    //returns the object of list which is present at the specified index
                    System.out.println("\n----------------------------------------");
                    System.out.println("\n\nOBJECT AT INDEX 2: "+ fruits.get(2));


              // removing all the elements of the ArrayList

              fruits.clear();

              System.out.println("\nARRAYLIST AFTER Clear(): "+ fruits);
          }
      }
```

RESULT: The above program is executed and obtained the output.

OUTPUT :

ORIGINAL LIST:

Pomegranate Lemon Avocado Orange Durian Watermelon

ARRAYLIST AFTER REMOVAL OF ELEMENTS:

Pomegranate Lemon Durian Watermelon

 ARRAYLIST AFTER UPDATION:

Pomegranate Lemon Durian Guava

 ARRAYLIST AFTER SORTING:

Durian Guava Lemon Pomegranate

Apple is in the List- true

Strawberry is in the List- false

SIZE OF THE ARRAYLIST: 4

OBJECT AT INDEX 2: Lemon

ARRAYLIST AFTER Clear(): []

Process finished with exit code 0

PROGRAM NO : 27

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

ALGORITHM :

Step 1: Start

Step 2: Declare a 2 D array named str of type String and read values into it.

Step 3: Create an object student of type LinkedList and put values in str into stud using add()

Step 4: Traverse through stud using Iterator and print the values.

Step 5: Stop.

PROGRAM CODE:

Main.java	
| | ```java
import java.util.*;

public class Main {

  public static void main(String[] args) {
    int n;
    String data;
    LinkedList<String> linkedList = new LinkedList<String>();
    System.out.println("Enter number of strings to be inserted : ");
    Scanner scanner = new Scanner(System.in);
    n = scanner.nextInt();
    System.out.println("Enter the string(s) : ");
    scanner.nextLine();
    for (int i = 0; i < n; i++) {
       data = scanner.nextLine();
       linkedList.add(data);
    }
    System.out.println("LinkedList: " + linkedList);
    System.out.println("Removing all the elements....");
    linkedList.clear();
    System.out.println(linkedList);
  }

}
``` |

RESULT: The above program is executed and obtained the output.

OUTPUT :

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
Enter number of strings to be inserted :
4
Enter the string(s) :
jabulani
Teamgeist
Telstar
Ferenova
LinkedList: [jabulani, Teamgeist, Telstar, Ferenova]
Removing all the elements....
[]

Process finished with exit code 0
```

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: Create an object 'fruits' of type Stack

Step 3: Read elements into fruits using add()

Step 4: Remove some elements using remove()

Step 5: Print the final stack

Step 6: Stop.

PROGRAM CODE:

Main.java	
| | ```java
import java.util.Scanner;
import java.util.Stack;

public class Main {

  public static void main(String[] args) {

    Stack<String> stack = new Stack<String>();
    Scanner scanner=new Scanner(System.in);
    System.out.println("Enter the size of Stack : ");
    int num=scanner.nextInt();

    for(int i =0;i<num;i++)
    {
      System.out.println("Add element for Stack : ");
      String str=scanner.next();
      stack.add(str);
    }

    System.out.println();
    System.out.println("Initial Stack: " + stack);
    System.out.println("Enter the position of element in stack to be
removed: ");
``` |

	int pos=scanner.nextInt();
	
	 String rem = stack.remove(pos);
	
	 System.out.println("\nRemoved element: "+ rem);
	 System.out.println("\nFinal Stack: " + stack);
	
	 }
	
	 }

RESULT: The above program is executed and obtained the output.

OUTPUT :

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
Enter the size of Stack :
4
Add element for Stack :
Blue
Add element for Stack :
Green
Add element for Stack :
Brown
Add element for Stack :
Violet

Initial Stack: [Blue, Green, Brown, Violet]
Enter the position of element in stack to be removed:
2

Removed element: Brown

Final Stack: [Blue, Green, Violet]

Process finished with exit code 0
```

PROGRAM NO : 29

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

ALGORITHM :

Step 1: Start

Step 2: Create an object 'stud' of type PriorityQueue. Step 3: Enter elements into stud using add().

Step 4: Remove some elements from stud using remove()

Step 5: Print the details with the help of Iterator

Step 6: Stop.

PROGRAM CODE:

Main.java	
| | ```java
import java.util.PriorityQueue;
import java.util.Scanner;

public class Main {

  public static void main(String[] args)
  {
    PriorityQueue<String> queue=new PriorityQueue<String>();
    Scanner scanner=new Scanner(System.in);
    System.out.println("Enter the size of the queue : ");
    int n=scanner.nextInt();
    System.out.println("Enter the elements : ");
    for(int i =0;i<n;i++)
    {
      String st=scanner.next();
      queue.add(st);

    }
    System.out.println("head:"+queue.element());
    System.out.println("head:"+queue.peek());
    System.out.println("Iterating the queue elements\n : ");
    for (String value : queue) {
      System.out.println(value);
    }
``` |

```
            queue.remove();
            queue.poll();
            System.out.println("After removing two elements \n");
            for (String s : queue) {
                System.out.println(s);
            }
        }

    }
```

RESULT: The above program is executed and obtained the output.

OUTPUT :

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
Enter the size of the queue :
4
Enter the elements :
Green
Blue
Red
Grey
head:Blue
head:Blue
Iterating the queue elements
 :
Blue
Green
Red
Grey
After removing two elements

Grey
Red

Process finished with exit code 0
```

PROGRAM NO : 30

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

ALGORITHM :

Step 1: Start

Step 2: Create a deque type object named 'd'.

Step 3: Put data into the d using appropriate functions.

Step 4: Remove the data using built in functions.

Step 5: Print the data in deque

Step 6: Stop.

PROGRAM CODE:

Main.java	
| | ```
import java.util.*;

public class Main {

  public static void main(String[] args) {
    int ch, data;
    Deque<Integer> dq = new LinkedList<Integer>();
    Scanner sc = new Scanner(System.in);
    do
    {
      System.out.println("\n_____MENU_____");
      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(1-6):");
      ch = sc.nextInt();
      sc.nextLine();
      switch(ch)
      {
``` |

```
                case 1: System.out.println("Enter the element to be inserted at
first:");
                    data = sc.nextInt();
                    dq.addFirst(data);
                    break;
                case 2: System.out.println("Enter the element to be inserted at
last:");
                    data = sc.nextInt();
                    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);
    }


}
```

RESULT: The above program is executed and obtained the output.

OUTPUT :

_____MENU_____

1.Insert the element at first

2.Insert the element at last

3.Delete the element at first

4.Delete the element at last

5.Display

6.Exit

Enter the choice(1-6):

1

Enter the element to be inserted at first:

34

_____MENU_____

1.Insert the element at first

2.Insert the element at last

3.Delete the element at first

4.Delete the element at last

5.Display

6.Exit

Enter the choice(1-6):

1

Enter the element to be inserted at first:

56

_____MENU_____

1.Insert the element at first

2.Insert the element at last

3.Delete the element at first

4.Delete the element at last

5.Display

6.Exit

Enter the choice(1-6):

1

Enter the element to be inserted at first:

100

_____MENU_____

1.Insert the element at first

2.Insert the element at last

3.Delete the element at first

4.Delete the element at last

5.Display

6.Exit

Enter the choice(1-6):

5

Elements:

[100, 56, 34]

_____MENU_____

1.Insert the element at first

2.Insert the element at last

3.Delete the element at first

4.Delete the element at last

5.Display

6.Exit

Enter the choice(1-6):

4

Element deleted from the last position

_____MENU_____

1.Insert the element at first

2.Insert the element at last

3.Delete the element at first

4.Delete the element at last

5.Display

6.Exit

Enter the choice(1-6):

5

Elements:

[100, 56]

_____MENU_____

1.Insert the element at first

2.Insert the element at last

3.Delete the element at first

4.Delete the element at last

5.Display

6.Exit

Enter the choice(1-6):

6

Process finished with exit code 0

PROGRAM NO : 31

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

ALGORITHM :

Step 1: Start

Step 2: Create a class named Book with data members id,name,author,publisher,quantity;

A parameterized constructor.

Step 3: Create a LinkedHashSet named Book and create an object named 'b' for the same.

Step 4: Create 3 objects b1,b2,b3 of type Book class

Step 5: Add the details in b1,b2,b3 into the object 'b' of type Book LinkedHashSet

Step 6: Print the elements of b using for loop

Step 7: Stop.

PROGRAM CODE:

Main.java	
| | ```
import java.util.LinkedHashSet;
public class Main {
public static void main(String[] args) {
LinkedHashSet<String> linkedset = new LinkedHashSet<String>();
// Adding element to LinkedHashSet
linkedset.add("Maruti");
linkedset.add("BMW");
linkedset.add("Honda");
linkedset.add("Audi");
linkedset.add("Maruti"); //This will not add new element as Maruti already
exists
linkedset.add("WalksWagon");
System.out.println("Size of LinkedHashSet=" + linkedset.size());
System.out.println("Original LinkedHashSet:" + linkedset);
System.out.println("Removing Audi from LinkedHashSet: " +
linkedset.remove("Audi"));
System.out.println("Trying to Remove Z which is not present: "
+ linkedset.remove("Z"));
System.out.println("Checking if Maruti is present=" +
linkedset.contains("Maruti"));
``` |

	System.out.println("Updated LinkedHashSet: " + linkedset); } }

RESULT: The above program is executed and obtained the output.

OUTPUT :

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
Size of LinkedHashSet=5
Original LinkedHashSet:[Maruti, BMW, Honda, Audi, WalksWagon]
Removing Audi from LinkedHashSet: true
Trying to Remove Z which is not present: false
Checking if Maruti is present=true
Updated LinkedHashSet: [Maruti, BMW, Honda, WalksWagon]

Process finished with exit code 0
```

PROGRAM NO : 32

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

ALGORITHM :

Step 1:Start

Step 2:Create an object named 'sone' of type HashSet.

Step 3:Add values into hashset using add() function.

Step 4:Create another object named 'stwo' of type HashSet

Step 5:Add values into hashset using add() function.

Step 6:Create another object named 'result_set' of type HashSet

Step 7:While traversing through the hashset using for loop, compare the two hashset objects

sone and stwo using contain() function and print the same.

Step 8:Stop.

PROGRAM CODE:

Main.java	
| | ```
import java.util.*;
public class Main {
  public static void main(String[] args) {
    int n;
    String str;
    HashSet<String> set1= new HashSet<String>();
    System.out.println("HashSet 1");
    System.out.println("Enter No. of countries:");
    Scanner sc=new Scanner(System.in);
    n=sc.nextInt();
    System.out.println("Enter the name of countries:");
    Scanner sc1=new Scanner(System.in);
    for(int i=0;i<n;i++) {
       str=sc1.nextLine();
       set1.add(str);
    }
    System.out.println("HashSet 2");
    HashSet<String> set2= new HashSet<String>();
    System.out.println("Enter No. of countries:");
``` |

```
                    n=sc.nextInt();
                    System.out.println("Enter the name of countries:");

                    for(int i=0;i<n;i++) {
                        str=sc1.nextLine();
                        set2.add(str);
                    }
                    System.out.println("Set1:"+set1);
                    System.out.println("Set2:"+set2);

                    HashSet<String> a= new HashSet<String>(set1);
                    a.addAll(set2);
                    System.out.println("Union of country set:"+a);

                    HashSet<String> b= new HashSet<String>(set1);
                    b.retainAll(set2);
                    System.out.println("Intersection of country set:"+b);

                    HashSet<String> c= new HashSet<String>(set1);
                    c.removeAll(set2);
                    System.out.println("Difference of country set:"+c);
                }
            }
```

RESULT: The above program is executed and obtained the output.

OUTPUT :

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
HashSet 1
Enter No. of countries:
4
Enter the name of countries:
England
Brazil
Argentina
Germany
HashSet 2
Enter No. of countries:
3
Enter the name of countries:
Brazil
Uruguay
Argentina
Set1:[Argentina, England, Brazil, Germany]
Set2:[Argentina, Uruguay, Brazil]
Union of country set:[Argentina, Uruguay, England, Brazil, Germany]
Intersection of country set:[Argentina, Brazil]
Difference of country set:[England, Germany]

Process finished with exit code 0
```

PROGRAM NO : 33

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

ALGORITHM :

Step 1:Start

Step 2:Create an object of type Map named 'map'

Step 3:Put values into map using put() funtion

Step 4:Using Map.Entry, traverse through map object and print the details suing getValue().

Step 5:Stop.

PROGRAM CODE:

Main.java	
| | ```
// Java program to illustrate HashMap class of java.util
// package

// Importing HashMap class
import java.util.HashMap;

// Main class
public class Main {

   // Main driver method
   public static void main(String[] args)
   {
      // Create an empty hash map by declaring object
      // of string and integer type
      HashMap<String, Integer> map = new HashMap<>();

      // Adding elements to the Map
      // using standard add() method
      map.put("Germany", 4);
      map.put("England", 1);
      map.put("Brazil", 5);

      // Print size and content of the Map
      System.out.println("Size of map is:- "
``` |

```
                + map.size());

        // Printing elements in object of Map
        System.out.println(map);

        // Checking if a key is present and if
        // present, print value by passing
        // random element
        if (map.containsKey("England")) {

            // Mapping
            Integer a = map.get("England");

            // Printing value fr the corresponding key
            System.out.println("value for key"
                    + " \"England\" is:- " + a);
        }
    }
}
```

RESULT: The above program is executed and obtained the output.

OUTPUT :

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
Size of map is:- 3
{England=1, Brazil=5, Germany=4}
value for key "England" is:- 1

Process finished with exit code 0
```

PROGRAM NO : 34

AIM: Program to Convert HashMap to TreeMap.

ALGORITHM :

Step 1:Start

Step 2:Create an object of type Map named 'map'

Step 3: Add values into map object

Step 4:To convert the Map type into TreeMap type, create an object of TreeMap type and

move all the values of map object using putAll() function.

Step 5:Print the values.

Step 6: Stop.

PROGRAM CODE:

Main.java	
| | ```java
import java.util.*;

public class Main {
  public static void main(String[] args) {
    // TODO Auto-generated method stub
    Map<Integer,String> hm=new LinkedHashMap<>();
    hm.put(1,"England");
    hm.put(1,"Spain");
    hm.put(2,"France");
    hm.put(5,"Brazil");
    hm.put(2,"Argentina");
    System.out.println("HashMap:"+hm);
    Map<Integer,String> tm=new TreeMap<>(hm);
    System.out.println("\n ******* Convert HashMap to TreeMap
*******\n");
    System.out.println("TreeMap:"+tm);
  }
}
``` |

RESULT: The above program is executed and obtained the output.

OUTPUT :

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
HashMap:{1=Spain, 2=Argentina, 5=Brazil}

 ******* Convert HashMap to TreeMap *******

TreeMap:{1=Spain, 2=Argentina, 5=Brazil}

Process finished with exit code 0
```

PROGRAM NO : 35

AIM: Program to draw Circle, Rectangle, Line in Applet.

ALGORITHM :

Step.1: Start the program.

Step.2: Define a class 'Main' that extends Applet class.

Step.3: Draw a line, rectangle and circle using drawLine, drawRect and drawOval methods of Graphics class respectively.

Step.4: Stop the program.

PROGRAM CODE:

Drawing.java	
| | ```
import java.applet.Applet;
import java.awt.*;

class Drawing extends Applet{

public void paint(Graphics g){
g.setColor(Color.green);

g.drawLine(20,30,20,300);
g.fillRect(70,100,30,30);
g.fillOval(70,200,30,30);

}
}
``` |
| Index.html | |
| | ```
<html>
<body>
<applet code="Drawing.class" width="300" height="300">
</applet>
</body>
</html>
``` |

RESULT: The above program is executed and obtained the output.

OUTPUT:

PROGRAM NO : 36

AIM: To write a program to find maximum of three numbers using AWT.

ALGORITHM :

Step.1: Start the program.

Step.2: Define a class 'Main' that extends Applet class and implements ActionListener interface.

Step.3: Using TextField class object, construct the required no. of textfields wide enough to hold the values entered by the user.

Step.4: Using Button class object, construct a labeled button that sends an instance of ActionEvent.

Step.5: Call addActionListener() method to send events from the button to the new listener.

Step.6: Get the string values from textfields and then parse them as integers.

Step.7: Compare each value using if-else statements to find the maximum value and set the result accordingly.

Step.8: Stop the program.

PROGRAM CODE:

Main.java	
| | ```
import java.applet.*;
    import java.awt.*;
    import java.awt.Graphics;
    import java.awt.event.*;

public class Main extends Applet implements ActionListener {
   Label l1,l2,l3,l4,l5,l6;
   TextField t1,t2,t3,t4,t5,t6;
   Button b;
   public void init(){
      l1 = new Label("MARK 1:");
      t1 = new TextField();
      l2 = new Label("MARK 2:");
      t2 = new TextField();
      l3 = new Label("MARK 3:");
      t3 = new TextField();
      l4 = new Label("MARK 4:");
``` |

```
t4 = new TextField();
l5 = new Label("MARK 5:");
t5 = new TextField();
l6 = new Label("PERCENTAGE:");
t6 = new TextField();
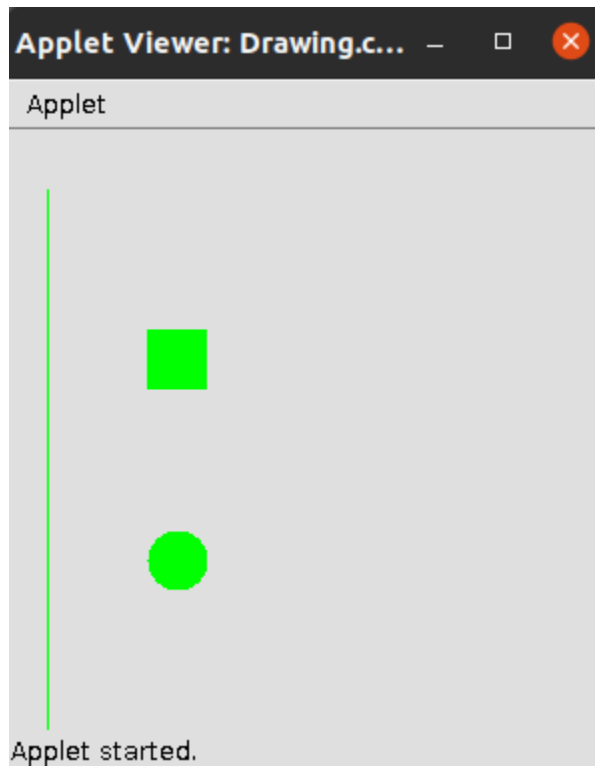

b = new Button("SEE STATUS");

setLayout(null);

l1.setBounds(450,50,70,20);
t1.setBounds(520,50,100,20);
l2.setBounds(450,80,70,20);
t2.setBounds(520,80,100,20);
l3.setBounds(450,110,70,20);
t3.setBounds(520,110,100,20);
l4.setBounds(450,140,70,20);
t4.setBounds(520,140,100,20);
l5.setBounds(450,170,70,20);
t5.setBounds(520,170,100,20);
l6.setBounds(450,200,100,20);
t6.setBounds(550,200,100,20);

b.setBounds(450,290,80,30);

add(l1);
add(l2);
add(l3);
add(l4);
add(l5);
add(l6);
add(t1);
add(t2);
add(t3);
add(t4);
add(t5);
add(t6);
add(b);
b.addActionListener(this);

}
public void actionPerformed(ActionEvent e){
    float m1, m2,m3, m4,m5,percent;

    m1= Float.parseFloat(t1.getText());
```

| | ```java
m2= Float.parseFloat(t2.getText());
m3= Float.parseFloat(t3.getText());
m4= Float.parseFloat(t4.getText());
m5= Float.parseFloat(t5.getText());

percent=((m1+m2+m3+m4+m5)*100)/500;

t6.setText(String.valueOf(percent));
repaint();
}


public void paint(Graphics g){

float p;
p= Float.parseFloat(t6.getText());

if(p> 50.0) {
g.setColor(Color.YELLOW);
g.fillOval(0,0,100,100);
g.setColor(Color.black);
g.fillOval(25,25,10,10);
g.fillOval(65,25,10,10);
g.setColor(Color.black);
g.fillArc (25,35,50,50,0,-180);
}
else {
g.setColor(Color.YELLOW);
g.fillOval(0,0,100,100);
g.setColor(Color.black);
g.fillOval(25,25,10,10);
g.fillOval(75,25,10,10);
g.setColor(Color.black);
g.drawArc(25,35,50,50,0,180);

}
}
}
``` |
| index.html | ```html
<html>
<body>
<applet code="Main.class" width="300" height="300">
</applet>
</body>
</html>
``` |


RESULT: The above program is executed and obtained the output.

OUTPUT:

PROGRAM NO: 37

AIM:

To find the percentage of marks obtained by a student in 5 subjects. Display a happy face ifhe secures above 50% or a sad face if otherwise.

ALGORITHM:

Step.1: Start the program.

Step.2: Define a class 'Face' that extends Applet class and implements ActionListenerinterface.

Step.3: Using TextField class object, construct textfields to receive marks of 5 subjects fromthe user.

Step.4: Using Button class object, construct a labeled button that sends an instance ofActionEvent.

Step.5: Call addActionListener() method to send events from the button to the new listener.

Step.6: Get the string values from textfields and then parse them as float values.

Step.7: Calculate the percentage: Percent = ((mark1+mark2+mark3+mark4+mark5)*100)/500
Step.8: Define a paint() method that contains functions from Graphics class to display ahappy face if student secures above 50% or a sad face if otherwise

Step.9: Stop the program.

PROGRAM CODE:

Main.java	
| | ```
import java.applet.*;
    import java.awt.*;
    import java.awt.Graphics;
    import java.awt.event.*;

public class Main extends Applet implements ActionListener {
   Label l1,l2,l3,l4,l5,l6;
   TextField t1,t2,t3,t4,t5,t6;
   Button b;
   public void init(){
      l1 = new Label("MARK 1:");
``` |

```
t1 = new TextField();
l2 = new Label("MARK 2:");
t2 = new TextField();
l3 = new Label("MARK 3:");
t3 = new TextField();
l4 = new Label("MARK 4:");
t4 = new TextField();
l5 = new Label("MARK 5:");
t5 = new TextField();
l6 = new Label("PERCENTAGE:");
t6 = new TextField();


b = new Button("SEE STATUS");

setLayout(null);

l1.setBounds(450,50,70,20);
t1.setBounds(520,50,100,20);
l2.setBounds(450,80,70,20);
t2.setBounds(520,80,100,20);
l3.setBounds(450,110,70,20);
t3.setBounds(520,110,100,20);
l4.setBounds(450,140,70,20);
t4.setBounds(520,140,100,20);
l5.setBounds(450,170,70,20);
t5.setBounds(520,170,100,20);
l6.setBounds(450,200,100,20);
t6.setBounds(550,200,100,20);

b.setBounds(450,290,80,30);

add(l1);
add(l2);
add(l3);
add(l4);
add(l5);
add(l6);
add(t1);
add(t2);
add(t3);
add(t4);
add(t5);
add(t6);
add(b);
b.addActionListener(this);
```

105

```
     }
   public void actionPerformed(ActionEvent e){
     float m1, m2,m3, m4,m5,percent;

     m1= Float.parseFloat(t1.getText());
     m2= Float.parseFloat(t2.getText());
     m3= Float.parseFloat(t3.getText());
     m4= Float.parseFloat(t4.getText());
     m5= Float.parseFloat(t5.getText());

     percent=((m1+m2+m3+m4+m5)*100)/500;

     t6.setText(String.valueOf(percent));
     repaint();
   }


   public void paint(Graphics g){

     float p;
     p= Float.parseFloat(t6.getText());

     if(p> 50.0) {
        g.setColor(Color.YELLOW);
        g.fillOval(0,0,100,100);
        g.setColor(Color.black);
        g.fillOval(25,25,10,10);
        g.fillOval(65,25,10,10);
        g.setColor(Color.black);
        g.fillArc (25,35,50,50,0,-180);
     }
     else {
        g.setColor(Color.YELLOW);
        g.fillOval(0,0,100,100);
        g.setColor(Color.black);
        g.fillOval(25,25,10,10);
        g.fillOval(75,25,10,10);
        g.setColor(Color.black);
        g.drawArc(25,35,50,50,0,180);

     }
   }
}
```

RESULT: The above program is executed and obtained the output.

OUTPUT:

PROGRAM NO : 38

AIM: Using 2D graphics commands in an Applet, construct a house. On mouse click event, change the color of the door from blue to red.

ALGORITHM:

Step.1: Start the program.

Step.2: Define a class 'House' that extends Applet and implements MouseListener.

Step.3: Define methods to add MouseListener to the panel.

Step.4: Using getX() and getY() methods, get the coordinates of the door to repaint when the

MousePressed event occurs.

Step.5: Stop the program.

PROGRAM CODE:

| House.java | ```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;

public class House extends Applet implements MouseListener
{
   int a,b;
   public void init()
   {
      addMouseListener( this);
   }
   public void paint(Graphics g)
   {
      int x[]={150,300,225};
      int y[]={150,150,25};
      g.drawPolygon(x,y,3);
      g.setColor(Color.GRAY);
      g.fillPolygon(x,y,3);

      g.drawRect(150,150,150,200);//House
      g.setColor(Color.CYAN);
``` |

```java
      g.fillRect(150,150,150,200);

      g.drawRect(200,200,50,150);//Door
      g.setColor(Color.blue);
      g.fillRect(200,200,50,150);

      if(a>200 && a<300 && b>200 && b<300)
      {
        g.setColor(Color.red);
        g.fillRect(200, 200, 50, 150);
      }
    }
    public void mouseClicked(MouseEvent e)
    {

    }
    public void mouseEntered(MouseEvent e)
    {

    }

    @Override
    public void mouseExited(MouseEvent e) {

    }

    public void mousePressed(MouseEvent e)
    {
      a=e.getX();
      b=e.getY();
      repaint();

    }
    public void mouseReleased(MouseEvent e)
    {

    }

}
```
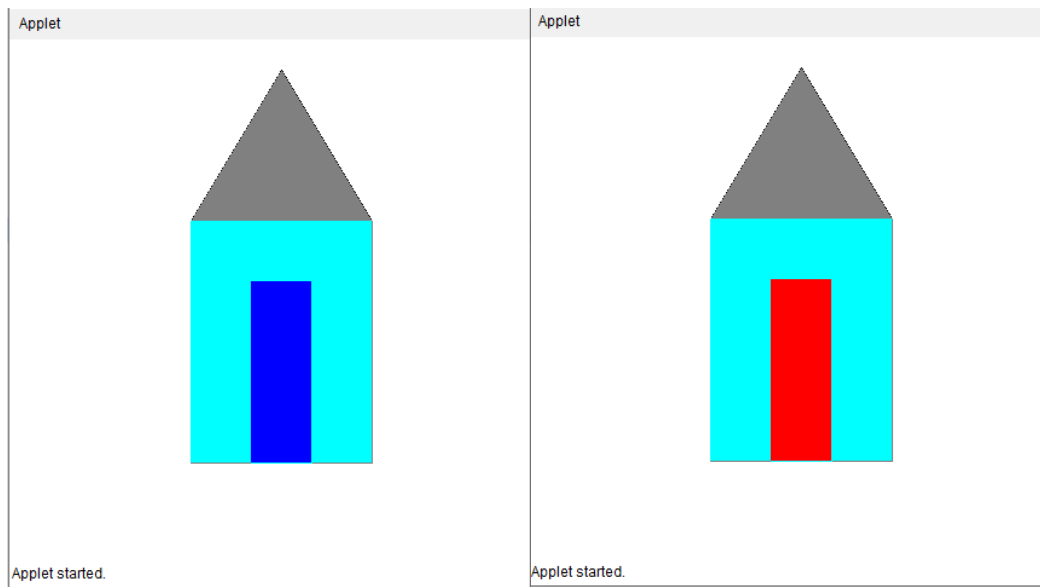
Index.html	

	`<html>`
	`<body>`
	`<applet code="House.class" width="600" height="600">`
	`</applet>`
	`</body>`
	`</html>`

RESULT: The above program is executed and obtained the output.

OUTPUT:

PROGRAM NO : 39

AIM: To implement a simple calculator using AWT components.

ALGORITHM:

Step.1: Start the program.

Step.2: Define a class 'calculator' that extends Frame and implements ActionListener interface.

Step.3: Using TextField class object, construct the required no. of textfields wide enough to hold the values entered by the user.

Step.4: Using Label class object, construct and provide the appropriate labels.

Step.5: Using Button class object, construct labeled buttons that send the instances of ActionEvent.

Step.6: Call addActionListener() method to send events from the button to the new listener.

Step.7: Get the string values from textfields and then parse them as integers.

Step.8: Perform various methods to add, subtract, multiply and divide those integers.

Step.9: Stop the program.

PROGRAM CODE:

MyCalculator.java	
| | ```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class MyCalculator extends Applet implements ActionListener
{
   TextField inp;
   public void init()
   {
        setBackground(Color.white);
        setLayout(null);
        int i;
        inp = new TextField();
        inp.setBounds(150,100,270,50);
``` |

```java
                this.add(inp);
                Button button[] = new Button[10];
                for(i=0;i<10;i++)
                {
                    button[i] = new Button(String.valueOf(9-i));
                    button[i].setBounds(150+((i%3)*50),150+((i/3)*50),50,50);
                    this.add(button[i]);
                    button[i].addActionListener(this);
                }
                Button dec=new Button(".");
                dec.setBounds(200,300,50,50);
                this.add(dec);
                dec.addActionListener(this);

                Button clr=new Button("C");
                clr.setBounds(250,300,50,50);
                this.add(clr);
                clr.addActionListener(this);

                Button operator[] = new Button[5];
                operator[0]=new Button("/");
                operator[1]=new Button("*");
                operator[2]=new Button("-");
                operator[3]=new Button("+");
                operator[4]=new Button("=");
                for(i=0;i<4;i++)
                {
                    operator[i].setBounds(300,150+(i*50),50,50);
                    this.add(operator[i]);
                    operator[i].addActionListener(this);
                }
                operator[4].setBounds(350,150,70,200);
                this.add(operator[4]);
                operator[4].addActionListener(this);
        }
        String num1="";
        String op="";
        String num2="";
        //Function to calculate the expression
        public void actionPerformed(ActionEvent e)
        {
            String button = e.getActionCommand();
          char ch = button.charAt(0);
            if(ch>='0' && ch<='9'|| ch=='.')
            {
                if (!op.equals(""))
```

112

```
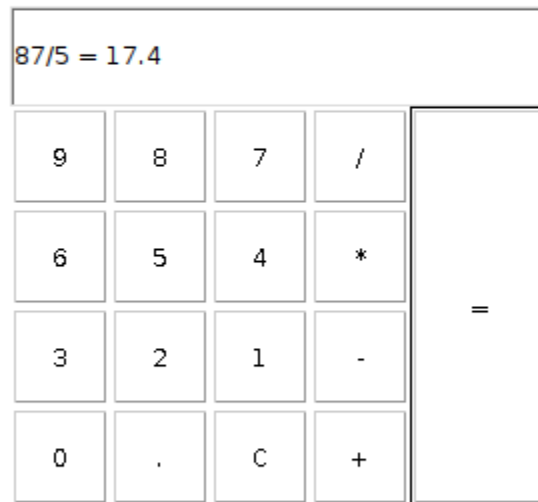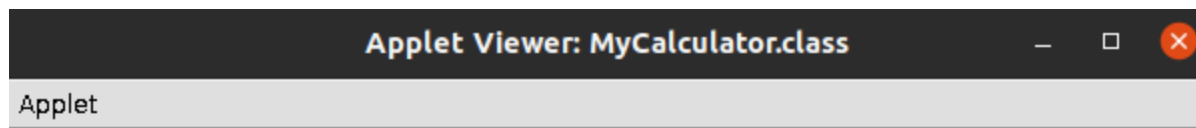                num2 = num2 + button;
            else
                num1 = num1 + button;
            inp.setText(num1+op+num2);
        }
        else if(ch=='C')
        {
            num1 = op = num2 = "";
            inp.setText("");
        }
        else if (ch =='=')
        {
            if(!num1.equals("") && !num2.equals(""))
            {
                double temp;
                double n1=Double.parseDouble(num1);
                double n2=Double.parseDouble(num2);
                if(n2==0 && op.equals("/"))
                {
                    inp.setText(num1+op+num2+" = Zero Division Error");
                    num1 = op = num2 = "";
                }
                else
                {
                    if (op.equals("+"))
                        temp = n1 + n2;
                    else if (op.equals("-"))
                        temp = n1 - n2;
                    else if (op.equals("/"))
                        temp = n1/n2;
                    else
                        temp = n1*n2;
                    inp.setText(num1+op+num2+" = "+temp);
                    num1 = Double.toString(temp);
                    op = num2 = "";
                }
            }
            else
            {
                num1 = op = num2 = "";
                inp.setText("");
            }
        }
        else
        {
            if (op.equals("") || num2.equals(""))
```

| | ```
                    op = button;
            else
            {
                double temp;
                double n1=Double.parseDouble(num1);
                double n2=Double.parseDouble(num2);
                if(n2==0 && op.equals("/"))
                {
                  inp.setText(num1+op+num2+" = Zero Division Error");
                  num1 = op = num2 = "";
                }
                else
                {
                  if (op.equals("+"))
                    temp = n1 + n2;
                  else if (op.equals("-"))
                    temp = n1 - n2;
                  else if (op.equals("/"))
                    temp = n1/n2;
                  else
                    temp = n1*n2;
                  num1 = Double.toString(temp);
                  op = button;
                  num2 = "";
                }
           }
              inp.setText(num1+op+num2);
        }
      }
}
``` |
| index.html | ```
<html>
<body>
<applet code="MyCalculator.class" width="600" height="600">
</applet>
</body>
</html>
``` |

RESULT: The above program is executed and obtained the output.

OUTPUT:

Applet Viewer: MyCalculator.class

Applet

```
87/5 = 17.4
```

9	8	7	/	
6	5	4	*	=
3	2	1	-	
0	.	C	+	

Applet started.

PROGRAM NO : 40

AIM: To develop a program that has a Choice component which contains the names of shapes such as rectangle, triangle, square and circle. Draw the corresponding shapes for given parameters as per user's choice.

ALGORITHM:

Step.1: Start the program.

Step.2: Define a class 'shapes' that extends Applet class and implements ItemListener interface.

Step.3: Declare a new constructor of the Choice class to create an empty Choice menu.

Step.4: Use add() method to include items in the menu.

Step.5: Using getSelectedItem() method, get the item chosen by the user and repaint accordingly.

Step.6: Stop the program.

PROGRAM CODE:

| Main.java | ```
import java.applet.*;
import java.awt.*;
import java.awt.Graphics;
import java.awt.event.*;
public class Main extends Applet implements ItemListener
  {
    Choice figure = new Choice();
    int Select;
   public void init()
   {

     figure.addItem("Select your choice");
     figure.addItem("Rectangle");
     figure.addItem("Square");
     figure.addItem("Circle");
     figure.addItem("Triangle");
     add(figure);
     figure.addItemListener(this);

``` |
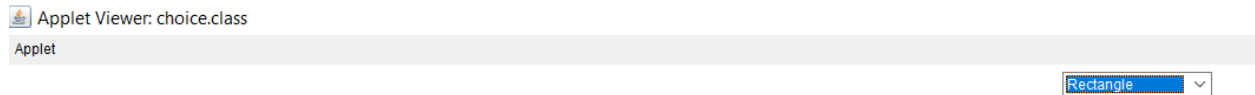
```
       }
     public void itemStateChanged (ItemEvent e)
   {

      Select = figure.getSelectedIndex();
      repaint();

   }

 public void paint(Graphics g)
       {

g.setColor(Color.red);
     super.paint(g);

     if (Select == 1)
     {

        g.drawRect(280, 100, 160,40);
     }
     if (Select == 2)
     {
        g.drawRect(50,50,100,100);
     }
     if (Select == 3)
     {

        g.drawOval(150,150,100,100);
     }
     if (Select ==4)
     {
        g.drawLine(120, 130, 280, 130);
        g.drawLine(120, 130, 200, 65);
        g.drawLine(200, 65, 280, 130);
     }
       }
}
```

index.html

```
<html>
<body>
<applet code="Main.class" width="600" height="600">
</applet>
</body>
</html>
```

RESULT: The above program is executed and obtained the output.

OUTPUT:

PROGRAM NO : 41

AIM: To develop a program to handle all mouse events and window events.

ALGORITHM:

Step 1: Start.

Step.1: Start the program.

Step.2: Define a class MouseDemo that extends Applet class and implements MouseListener interface.

Step.3: Define methods to add MouseListener to the panel which will have the following methods:

- void mouseClicked(MouseEvent me) - Invoked when the mouse has been clicked.
- void mousePressed(MouseEvent me) - Invoked when the mouse has been pressed.
- void mouseReleased(MouseEvent me) - Invoked when the mouse has been released.
- void mouseEntered(MouseEvent me) - Invoked when the mouse has entered the panel.
- void mouseExited(MouseEvent me) - Invoked when the mouse has exited the panel.
- void mouseDragged(MouseEvent me) - Invoked when the mouse has been dragged.

Step.4: Using getX() and getY() methods, get the location (or movements) of mouse pointer on the panel. Use them to display the necessary message in the output.

Step.5: Define another class WindowEvents that extends Applet class and implements

WindowListener interface.

Step.6: Define methods to add WindowListener to the panel which will have the following methods:

- void windowActivated(WindowEvent arg0) - Invoked when the window has been activated.
- void windowOpened(WindowEvent arg0) - Invoked when the window has been Opened.
- void windowDeactivated(WindowEvent arg0) - Invoked when the window has been deactivated.
- void windowIconified(WindowEvent arg0) - Invoked when the window has been iconified.
- void windowDeiconified(WindowEvent arg0) - Invoked when the window has been deiconified
- void windowClosed(WindowEvent arg0) - Invoked when the window has been closed.

Step.7: Display the appropriate message in the output.
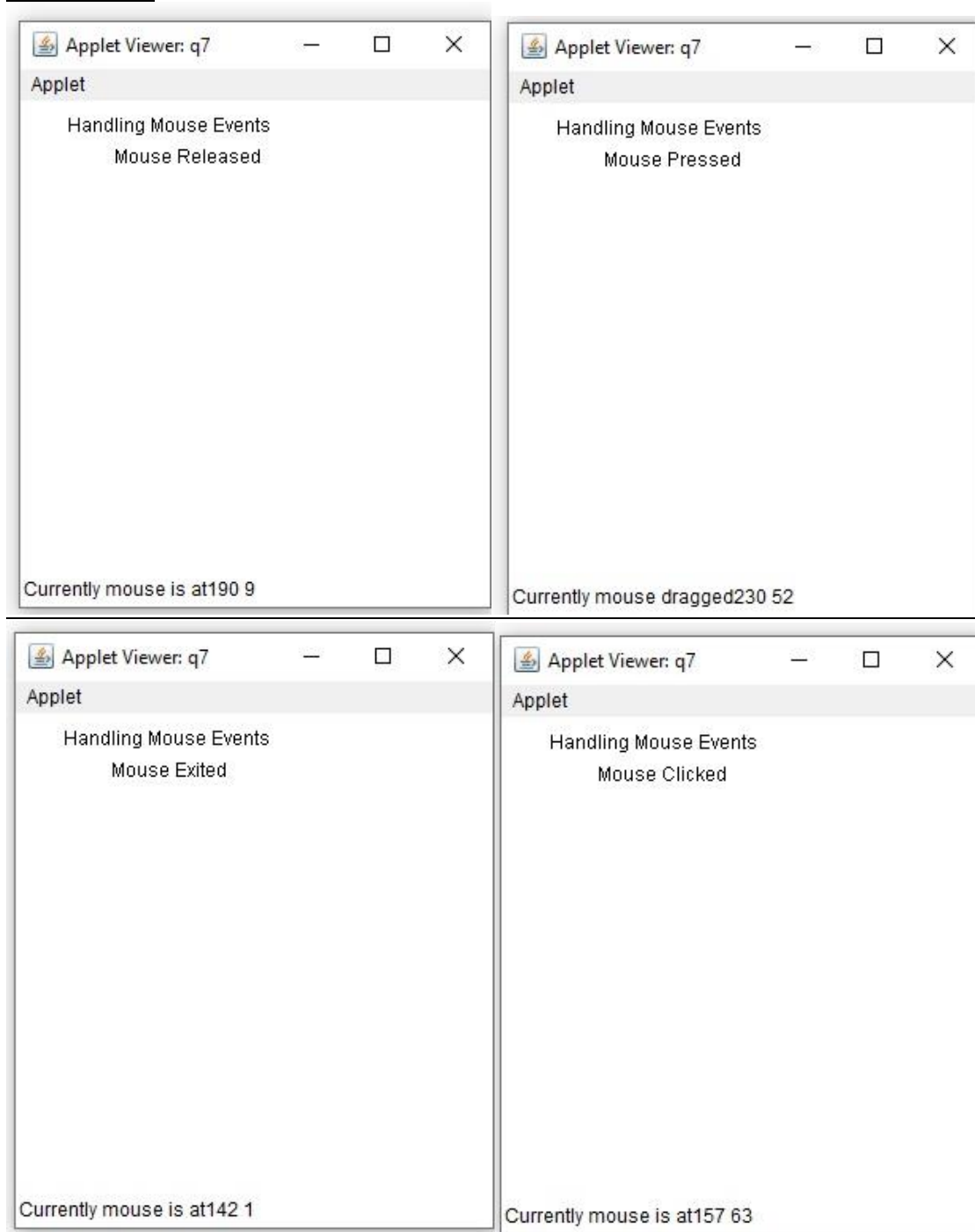
Step.8: Stop the program.
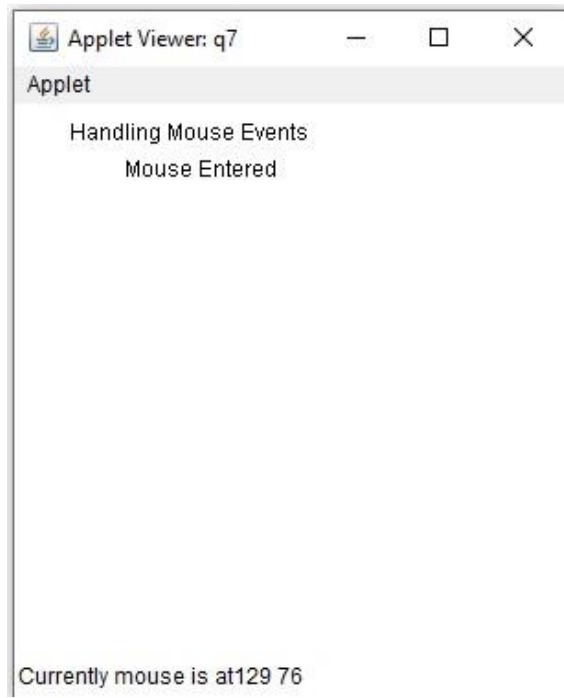
PROGRAM CODE:

Main.java	
| | ```java
import java.awt.*;
import java.applet.*;
import java.awt.event.*;


public class Main extends Applet implements
MouseListener,MouseMotionListener
{
int mx=0;
int my=0;
String msg="";
public void init()
{
addMouseListener(this);
addMouseMotionListener(this);
}
public void mouseClicked(MouseEvent me)
{
mx=20;
my=40;
msg="Mouse Clicked";
repaint();
}
public void mousePressed(MouseEvent me)
{
mx=30;
my=60;
msg="Mouse Pressed";

repaint();
}
public void mouseReleased(MouseEvent me)
{
mx=30;
my=60;
msg="Mouse Released";
repaint();
}
public void mouseEntered(MouseEvent me)
``` |

| | ```
{
mx=40;
my=80;
msg="Mouse Entered";
repaint();
}
public void mouseExited(MouseEvent me)
{
mx=40;
my=80;
msg="Mouse Exited";
repaint();
}
public void mouseDragged(MouseEvent me)
{
mx=me.getX();
my=me.getY();
showStatus("Currently mouse dragged"+mx+" "+my);

repaint(); }
public void mouseMoved(MouseEvent me)
{
mx=me.getX();
my=me.getY();
showStatus("Currently mouse is at"+mx+" "+my);
repaint();
}
public void paint(Graphics g)
{
g.drawString("Handling Mouse Events",30,20);
g.drawString(msg,60,40);
}
}
``` |
| index.html | ```
<html>
<body>
<applet code="Main.class" width="600" height="600">
</applet>
</body>
</html>
``` |

RESULT: The above program is executed and obtained the output.

OUTPUT:

Applet Viewer: q7

Applet

Handling Mouse Events
Mouse Released

Currently mouse is at190 9

Applet Viewer: q7

Applet

Handling Mouse Events
Mouse Pressed

Currently mouse dragged230 52

Applet Viewer: q7

Applet

Handling Mouse Events
Mouse Exited

Currently mouse is at142 1

Applet Viewer: q7

Applet

Handling Mouse Events
Mouse Clicked

Currently mouse is at157 63

PROGRAM NO : 42

AIM: To develop a program to handle Key events.

ALGORITHM:

Step.1: Start the program.

Step.2: Define a class keys that extends Applet and implements KeyListener.

Step.3: Define methods to add KeyListener to the panel which will have the following methods:

- void keyTyped(KeyEvent e) – Invoked when a key has been typed.
- void keyPressed(KeyEvent e) - Invoked when a key has been pressed.
- void keyReleased(KeyEvent e) - Invoked when a key has been released.

Step.4: Using getKeyChar(), get the unicode and character representation of the key pressed. Use them to display the necessary message in the output.

Step.5: Stop the program.

PROGRAM CODE:

| Main.java | ```java
import java.awt.*;
import java.awt.event.*;
import java.applet.*;

public class Main extends Applet implements KeyListener
{
String msg="Typed";
int x=30,y=50;
public void init()
{
addKeyListener(this);
requestFocus();
}
public void keyTyped(KeyEvent ke)
{
msg+=ke.getKeyChar();
repaint();
}
public void keyReleased(KeyEvent ke)
``` |
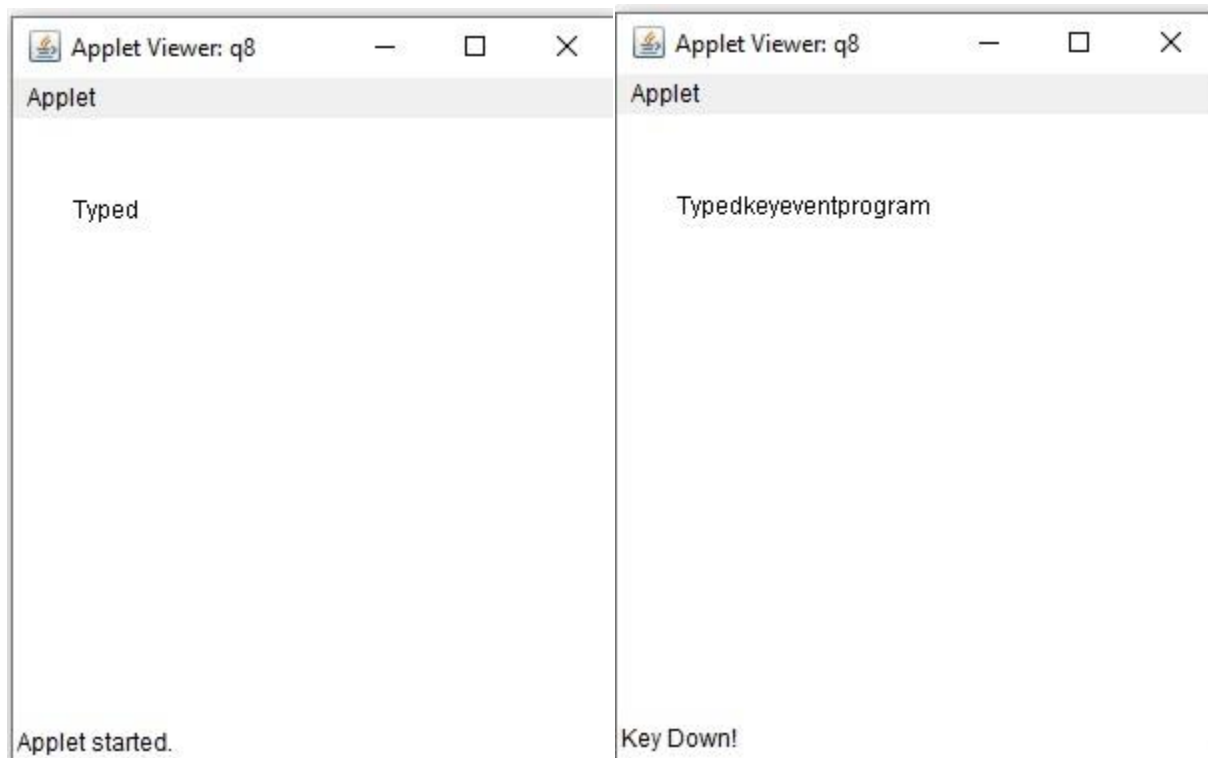
124

```
{
showStatus("Key Up!");
}
public void keyPressed(KeyEvent ke)
{
showStatus("Key Down!");
}
public void paint(Graphics G)
{
G.drawString(msg,x,y);
}
}
```

RESULT: The above program is executed and obtained the output.

OUTPUT:

PROGRAM NO : 43

AIM: To write program to list the sub directories and files in a given directory and also search for a file name.

ALGORITHM:

Step.1: Start the program.

Step.2: Create a class named 'FilesList' that implements FilenameFilter interface.

Step.3: Create an object for the class File to to initialize its constructor with the file source.

Step.4: Using list(), get the names of all the files present in the directory.

Step.5: Create an object for the FileNameFilter interface that contains the method Boolean accept (File dir, String name) to test if a specified file should be included in the file list or not. .

Step.6: Filter accordingly and store the file names to the list.

Step.7: Display the list.

Step.8: Stop the program.

PROGRAM CODE:

Main.java	
| | ```java
import java.io.*;
public class Main {

public static void main(String[] args) {


    File file = new File("../../../");


    String[] fileList = file.list();

    for(String str : fileList) {
       System.out.println(str);
    }
``` |

```
                    FilenameFilter filter = new FilenameFilter() {
                       public boolean accept (File dir, String name) {
                          return name.startsWith("A");
                       }
                    };
                    System.out.println("\n");
                    String[] children = file.list(filter);
                    if (children == null) {

                       System.out.println("Either dir does not exist or is not a directory");
                    } else {
                       for (int i = 0; i< children.length; i++) {
                          String filename = children[i];
                          System.out.println(filename);
                       }
                    }
                 }

          }
```

RESULT: The above program is executed and obtained the output.

OUTPUT:

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
$RECYCLE.BIN
Android
Android SDK
Config.Msi
machines                        .
notes
os pool
pic
SDK
setup
software
System Volume Information
workspace


Android
Android SDK
```

127

PROGRAM NO : 44

AIM: To write a program to write to a file, then read from the file and display the contents on the console.

ALGORITHM:

Step.1: Start the program.

Step.2: Create a class named 'Main'.

Step.3: Create an object of the class File to initialize its constructor with the file source.

Step.4: Create and use an object for the FileWriter class to write the file.

Step.5: Create and use an object for the BufferedReader class to read the stream of characters the specified file.

Step.6: Display the contents read from the file on the console.

Step.7: Stop the program.

PROGRAM CODE:

Main.java	
| | ```java
import java.io.BufferedReader;
    import java.io.FileReader;
    import java.io.FileWriter;
    import java.io.IOException;

public class Main {

   public static void main(String[] args) {

      try {
         FileWriter writer = new FileWriter("java_write.txt",true);
         writer.write("new file is created");
         writer.close();
         FileReader reader = new FileReader("java_write.txt");
         BufferedReader br= new BufferedReader(reader);
         String line;
         System.out.println("Data read from the file");
``` |
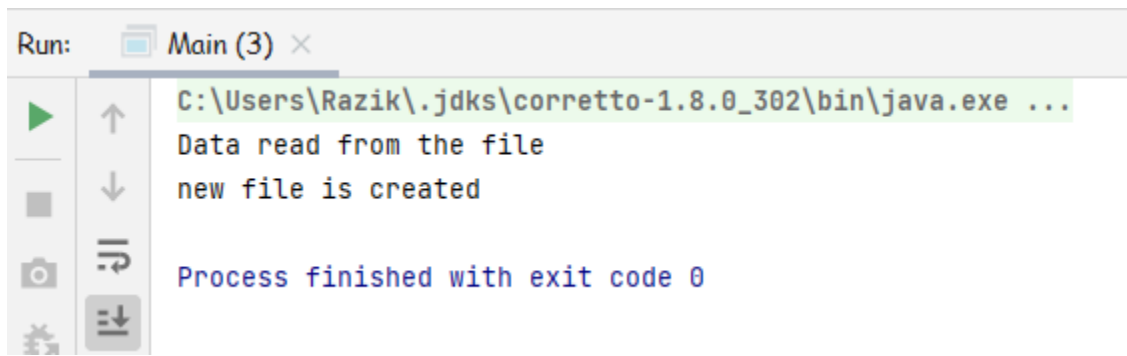
```
        while ((line = br.readLine()) != null) {
            System.out.println(line);
        }
        reader.close();

    } catch (IOException e) {
        System.out.println("-----Error-----");
    }

}

}
```

RESULT: The above program is executed and obtained the output.

OUTPUT:

```
Run:     Main (3) ×

    C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
    Data read from the file
    new file is created

    Process finished with exit code 0
```

PROGRAM NO : 45

AIM: To write a program to copy one file to another.

ALGORITHM:

Step.1: Start the program.

Step.2: Create a class named 'Main'.

Step.3: Create and use an object for the BufferedReader class to read the stream of characters from the specified file.

Step.4: Create and use an object for the FileWriter class to write the stream of characters read by the BufferedReader, to the file.

while ((s = br.readLine()) != null) {

fw.write(s);

}

Step.6: Display the appropriate message on the console.

Step.7: Stop the program.

PROGRAM CODE:

Main.java	
| | ```
import java.io.FileInputStream;
    import java.io.FileOutputStream;
    import java.io.IOException;


public class Main {

   public static void main(String[] args) throws IOException{
      // TODO Auto-generated method stub




      FileInputStream fileinput = new FileInputStream("source.txt");
``` |

```
            FileOutputStream fileoutput = new FileOutputStream("destination.txt");

        int i;
        while((i = fileinput.read()) != -1){
           fileoutput.write(i);
        }
        System.out.println("copied");
        fileinput.close();
        fileoutput.close();


     }
  }
```

RESULT: The above program is executed and obtained the output.

OUTPUT:

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
copied

Process finished with exit code 0
```

PROGRAM NO : 46

AIM: To write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files.

ALGORITHM:

Step.1: Start the program.

Step.2: Create a class named 'CopySep'.

Step.3: Create an object for the class File to initialize its constructor with the given file.

Step.4: Get user inputs via the console, for the integers to be inserted into the file.

Step.6: Using an object for the FileWriter class, write those integers into the file.

Step.7: Using objects for the FileOutputStream class, create two separate files to store even and odd integers respectively and copy the integers accordingly to separate files just created.

while((i=r.read()) != -1)

{

if(i%2==0) fo1.write(i);

else

fo2.write(i);

}

Step.8: Stop the program.

PROGRAM CODE:

Main.java	
| | ```java
import java.io.FileInputStream;
    import java.io.FileOutputStream;
    import java.io.IOException;

public class Main {

  public static void main(String[] args) throws IOException {
    // TODO Auto-generated method stub

    FileInputStream source = new FileInputStream ("source.txt");
    FileOutputStream destination_odd = new FileOutputStream ("odd.txt");
    FileOutputStream destination_even = new FileOutputStream
("even.txt");

    int i;
    while((i = source.read()) != -1){
      if(i%2==0) {

        destination_even.write(i);
      }
      else {

        destination_odd.write(i);
      }
    }
    System.out.println("copied");
    source.close();
    destination_even.close();
    destination_odd.close();
  }
}
``` |

RESULT: The above program is executed and obtained the output.

OUTPUT:

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
copied


Process finished with exit code 0
```

Main.java ×   even.txt ×   odd.txt ×

```
1    þemIsumissimyummyeoeiigayeseigiusy
2    oemIsumaseeeiusy'ssaaummyeeesicee15s
3    weaukowieookagaeyoyeascameiomakeayesecimeook
4    Iassuieooyieceuiesuasoeeaioeecoicyeseigemaiigesseiayucage
5    Iwasouaiseie19swieeeaseoeaseseescoaiigoemIsumassages
6    amoeeceywieskouisigsowaeikeAusageMakeicuigesiosooemIsum
```

Main.java ×   even.txt ×   odd.txt ×

```
1    Lr p  pl d txt f th prntn nd tpttn ndtr.
2    Lr p h bn th ndtr tndrd d txt vr n th 00,
3    hn n nnn prntr t  ll f tp nd rbld t t   tp pn b.
4    t h rvvd nt nl fv ntr, bt l th lp nt ltrn tpttn, rnn ntll nhnd.
5    t  pplrd n th 60 th th rl f Ltrt ht ntnn Lr p p,
6    nd r rntl th dtp pblhn ftr l ld Pr nldn vrn f Lr p.
```

134

PROGRAM NO : 47

AIM: To implement client server communication using Socket – TCP/IP.

ALGORITHM:

Step.1: Start the program.

Step.2: To create the Client application, create an instance of ClientSocket class.

2.1   : Initiate connection to the server using hostname and a port number.

2.2   : Send data to the server using an OutputStream object.

2.3   : Read data from the server using an InputStream object.

2.4   : Close the connection.

Step.3: To create the Server application, create an instance of ServerSocket class.

3.1   : Wait till a connection is established. Socket s = ss.accept();

3.2   : Receive data from the client using an InputStream object.

3.3   : Send data to the client using an OutputStream object.

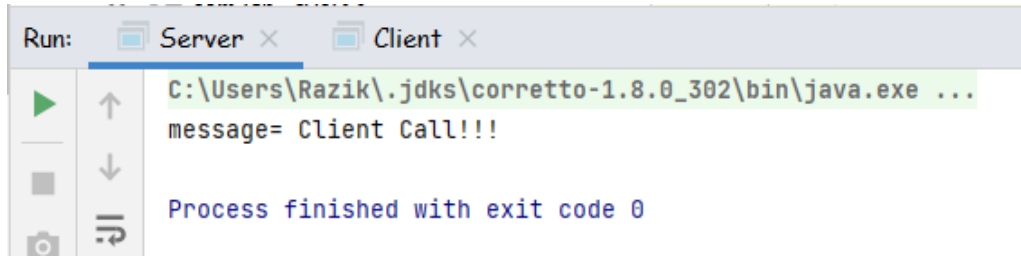3.4   : Close the connection.

Step.4: Stop the program.

PROGRAM CODE:

| Client.java | ```java
import java.io.*;
import java.net.*;

public class Client {
   public static void main(String[] args) {
      try{
         Socket socket=new Socket("localhost",7011);
         DataOutputStream dout=new
DataOutputStream(socket.getOutputStream());
         dout.writeUTF("Client Call!!!");
         dout.flush();
         dout.close();
         socket.close();
      }catch(Exception e){System.out.println(e);}
   }
}
``` |
| Server.java | ```java
import java.io.*;
import java.net.*;

public class Server {
   public static void main(String[] args){
      try{
         ServerSocket serverSocket=new ServerSocket(7011);
         Socket socket=serverSocket.accept();
         //establishes connection

         DataInputStream dis=new DataInputStream(socket.getInputStream());
         String  str=(String)dis.readUTF();
         System.out.println("message= "+str);
         serverSocket.close();
      }
      catch(Exception e){
         System.out.println(e);
      }
   }
}
``` |

RESULT: The above program is executed and obtained the output.

OUTPUT:

Run: Server ✕   Client ✕

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
message= Client Call!!!

Process finished with exit code 0
```

PROGRAM NO : 48

AIM: Client Server communication using DatagramSocket - UDP

ALGORITHM:

Step.1: Start the program.

Step.2: Create the Client application:

2.1   : Create a DatagramSocket object to carry the packet to the destination and to receive it whenever the server sends any data.

2.2   : Create the packet for sending/receiving data via a DatagramSocket.

DatagramPacket(byte buf[], int length, InetAddress inetaddress, int port):-

2.3   : Invoke a send() or receive() call on socket object.

2.4   : Close the connection.

Step.3: Create the Server application:

3.1   : Create a DatagramSocket object to listen at the port specified.

3.2   : Create the packet for sending/receiving data via a DatagramSocket.

3.3   : Invoke a send() or receive() call on socket object.

3.4   : Close the connection.

Step.4: Stop the program.

PROGRAM CODE:

| Client.java | ```java
import java.io.*;
import java.net.*;

public class Client {
    public static void main(String[] args) throws IOException {
        DatagramSocket client= new DatagramSocket();
        InetAddress add=InetAddress.getByName("localhost");
        String str ="Ping from Client!!!";
        byte[] bufBytes = str.getBytes();
        DatagramPacket datagramPacket=new
DatagramPacket(bufBytes,bufBytes.length,add,4220);
        client.send(datagramPacket);
        client.close();
    }
}
``` |
|---|---|
| Server.java | ```java
import java.io.*;
import java.net.*;

public class Server {
    public static void main(String[] args) throws IOException {
        DatagramSocket server=new DatagramSocket(4220);
        byte[] buf=new byte[256];
        DatagramPacket packet=new DatagramPacket(buf,buf.length);
        server.receive(packet);
        String response =new String(packet.getData());
        System.out.println(" Server : "+response);
        server.close();
    }
}
``` |

RESULT: The above program is executed and obtained the output.

OUTPUT:

Run: Server (1) ✕   Client (1) ✕

```
C:\Users\Razik\.jdks\corretto-1.8.0_302\bin\java.exe ...
 Server : Ping from Client!!!◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻◻

Process finished with exit code 0
```