DEPARTMENT OF COMPUTER APPLICATION

TKM COLLEGE OF ENGINEERING

KOLLAM – 691005

20MCA132 – OBJECT ORIENTED PROGRAMMING LAB

PRACTICAL RECORD BOOK

Second Semester MCA

2020-2021

Submitted by:

NAME: NITHIN RAJ

ROLL NO: 20MCA227

DEPARTMENT OF COMPUTER APPLICATION

TKM COLLEGE OF ENGINEERING

KOLLAM – 691005

Certificate

This is a bonafide record of the work done by NITHIN RAJ in the Second Semester in OBJECT ORIENTED PROGRAMMNG LAB Course (20MCA132) towards the partial fulfillment of the degree of Master of Computer Applications during the academic year 2020-2021.

Staff Member in-charge	Examiner
…………………………….	………………………..

INDEX

SL.NO	PROGRAMS	PAGE NO
	COURSE OUTCOME 1	
1	Creating Objects of a Class	1
2	Matrix Addition	3
3	Add Complex numbers	6
4	Check symmetric matrix or not	8
5	Inner and Nested Class	12
	COURSE OUTCOME 2	
6	Sort Strings	14
7	Search an element in array	16
8	String Manipulation	18
9	Array of Objects	20
	COURSE OUTCOME 3	
10	Overloaded functions	23
11	Single Inheritance	26
12	Multilevel Inheritance	29
13	Print details of Book using Inheritance	33
14	Display academic details of student	36
15	Menu driven program to find area and perimeter	38
16	Bill Preparation	41
	COURSE OUTCOME 4	
17	Graphics Package	44
18	Arithmetic Package	47
19	Username and password authentication	50
20	Average of N Positive Integers	53
21	Generate multiplication table using thread	56
22	Fibonacci using runnable interface	59
23	Producer/Consumer Problem	62
24	Generic Stack	66
25	Bubble Sort	68
26	Array List	70
27	Remove elements from Linked list	72
28	Remove an object from stack	74
29	Creation of queue	76
30	Addition and Deletion in Dequeue	78

31	Creation of Set	82
32	Compare two hast set	84
33	Working of Map Interface	86
34	Convert Hash Map to Tree Map	88
	COURSE OUTCOME 5	
35	Draw shapes in Applet	90
36	Find maximum using AWT	92
37	Display a happy face /sad face according to percentage	95
38	Mouse Events	99
39	Simple Calculator using AWT	102
40	Choice Components with shapes	110
41	Handling mouse events and window events	113
42	Handling Key Events	117
	COURSE OUTCOME 6	
43	Listing Sub directories and files	119
44	Read and Display File	121
45	Copy one file to another	123
46	Copy Even and Odd numbers to Separate Files	125
47	TCP socket programming	127
48	UDP socket programming	129

PROGRAM: 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.

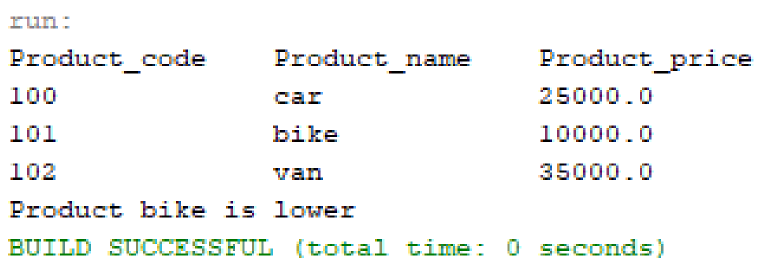
Step 5: Stop.

PROGRAM CODE:

product.java

```java
public class product {
    int pcode;
    String pname;
    double price;

    void data(int c, String n, double p)
    {
        pcode = c;
        pname = n;
        price = p;
    }
    void display()
    {
        System.out.println(pcode+"\t\t"+pname+"\t\t"+price);
    }
    void small(double a, double b, double c)
    {
        if(a<b && a<c)
        {
            System.out.println("Product car is lower");
        }
        else if(b<c && b<a)
        {
            System.out.println("Product bike is lower");
        }
        else
        {
            System.out.println("Product van is lower");
        }
    }
    public static void main(String[] args) {
        product obj1 = new product();
        product obj2 = new product();
        product obj3 = new product();
        obj1.data(100,"car",25000);
        obj2.data(101,"bike",10000);
        obj3.data(102,"van",35000);
        System.out.println("Product_code\tProduct_name\tProduct_price");
        obj1.display();
        obj2.display();
        obj3.display();
        obj1.small(obj1.price,obj2.price,obj3.price);

    }
}
```

OUTPUT:



```
run:
Product_code    Product_name    Product_price
100             car             25000.0
101             bike            10000.0
102             van             35000.0
Product bike is lower
BUILD SUCCESSFUL (total time: 0 seconds)
```

PROGRAM: 2

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

ALGORITHM :

Step 1: Start.

Step 2: Define a class having name matrix_add.

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[][].

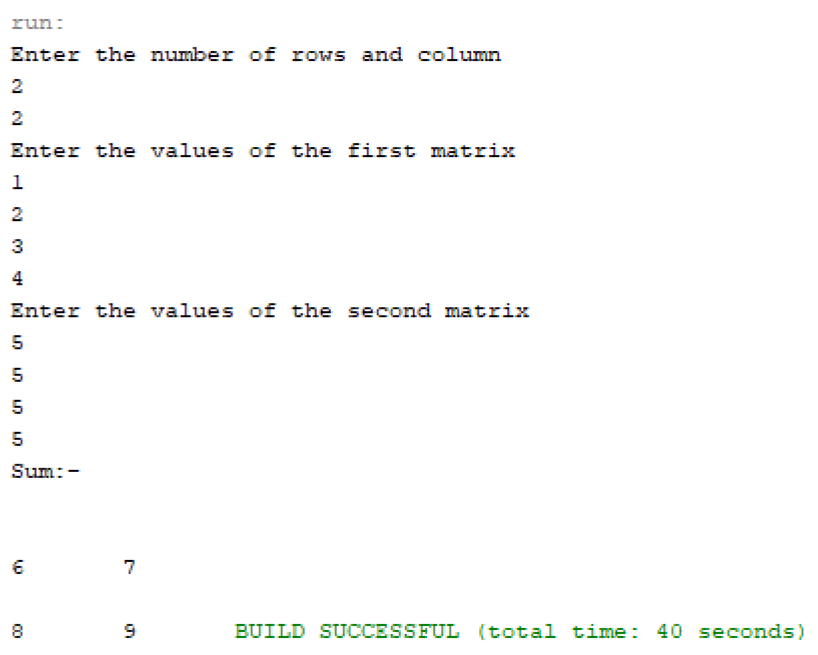
```
Step 5: Add the two matrices using for loop.
        for i=0 to i<m
        for j=0 to j<n
        mat1[i][j] + mat2[i][j] and store it in to the matrix res[i][j] .
Step 6: Print sum of matrices res[i][j].
Stop 7: Stop.
```

PROGRAM CODE:

matrix_add.java	
	```
	import java.util.Scanner;


	public class matrix_add {

	    public static void main(String[] args) {
	        int m,n;
	        int mat1[][]= new int[3][3];
	        int mat2[][]= new int[3][3];
	        int sum[][] = new int[3][3];
	        System.out.println("Enter the number of rows and column");
	        Scanner sc = new Scanner(System.in);
	        m = sc.nextInt();
	        n = sc.nextInt();
	        System.out.println("Enter the values of the first matrix");
	        for(int i=0;i<m;i++)
	        {
	            for(int j=0;j<n;j++)
	            {
	```

```
              mat1[i][j] = sc.nextInt();
           }
        }

        System.out.println("Enter the values of the second matrix");
        for(int i=0;i<m;i++)
        {
           for(int j=0;j<n;j++)
           {
              mat2[i][j] = sc.nextInt();
           }
        }

        for(int i=0;i<m;i++)
        {
           for(int j=0;j<n;j++)
           {
              sum[i][j] = mat1[i][j]+mat2[i][j];
           }
        }
        System.out.println("Sum:-");
        for(int i=0;i<m;i++)
        {
           System.out.println("\n");
           for(int j=0;j<n;j++)
           {
               System.out.print(sum[i][j]+"\t");

           }
        }
    }
}
```

OUTPUT:



```
run:
Enter the number of rows and column
2
2
Enter the values of the first matrix
1
2
3
4
Enter the values of the second matrix
5
5
5
5
Sum:-


6       7

8       9       BUILD SUCCESSFUL (total time: 40 seconds)
```

PROGRAM: 3

AIM: Add complex numbers.

ALGORITHM:

Step 1: Start.

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

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

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

Step 5: Print the sum value.

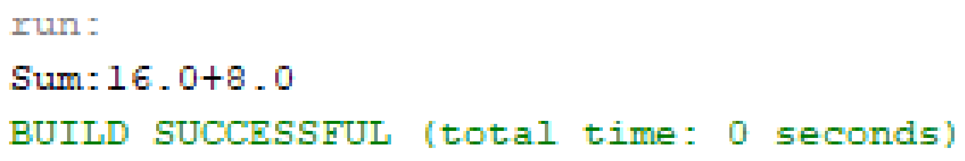
Step 6: Stop.

PROGRAM CODE:

complex.java

```java
public class complex {

    double real,img;

    complex(double a,double b)

    {

        real = a;

        img = b;

    }

    public static complex sum(complex c1,complex c2)

    {

        complex t = new complex(0,0);

        t.real = c1.real + c2.real;

        t.img = c1.img + c2.img;
        return t;

    }

    public static void main(String[] args) {

        complex c1 = new complex(12,3);

        complex c2 = new complex(4,5);

        complex add = sum(c1,c2);

        System.out.println("Sum:"+add.real+"+"+add.img);

    }

}
```

OUTPUT:



```
run:
Sum:16.0+8.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

PROGRAM: 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.
```

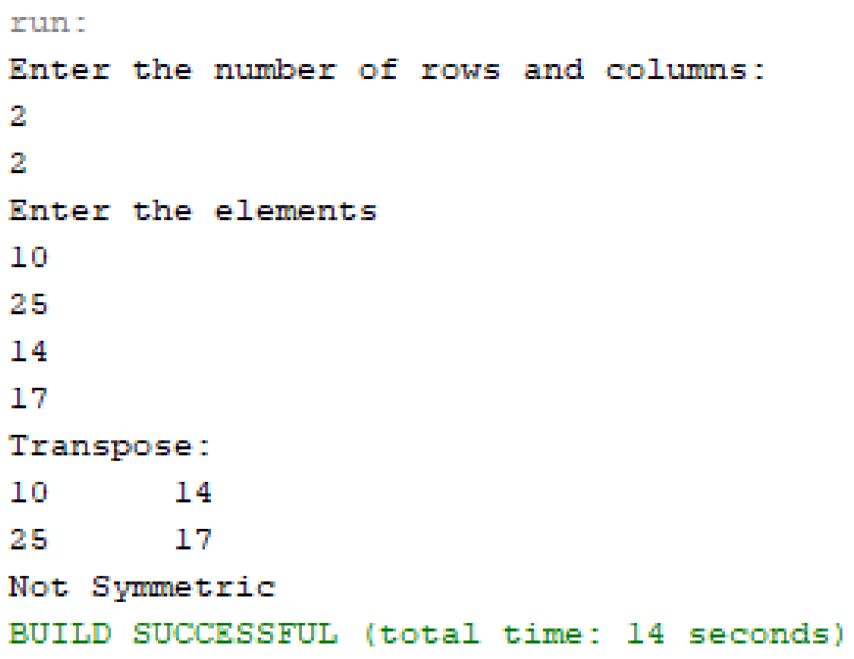
PROGRAM CODE:

symmetric.java	
	```
	import java.util.Scanner;



	public class symmetric {

	    public static void main(String[] args) {

	        int r,c;

	        int flag=0;

	        Scanner sc = new Scanner(System.in);

	        System.out.println("Enter the number of rows and columns:");

	        r = sc.nextInt();

	        c = sc.nextInt();
	```

```
int a[][] = new int[r][c];

int b[][] = new int[r][c];

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

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

{

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

    {

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

    }

}

if(r==c)

{

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

    {

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

        {

            b[i][j] = a[j][i];

        }

    }

}

System.out.println("Transpose:");

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

{

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

    {
```

```
                System.out.print(b[i][j]+"\t");
            }
            System.out.println("");
        }
        for(int i=0;i<r;i++)
        {
            for(int j=0;j<c;j++)
            {
               if(a[i][j]!=b[i][j])
                  {
                        flag = 1;
                        break;
                  }
            }
        }
        if(flag==0)
        {
            System.out.println("Symmetric");
        }
        else
        {
            System.out.println("Not Symmetric");
        }
    }
}
```

OUTPUT:


```
run:
Enter the number of rows and columns:
2
2
Enter the elements
10
25
14
17
Transpose:
10      14
25      17
Not Symmetric
BUILD SUCCESSFUL (total time: 14 seconds)
```

PROGRAM: 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.

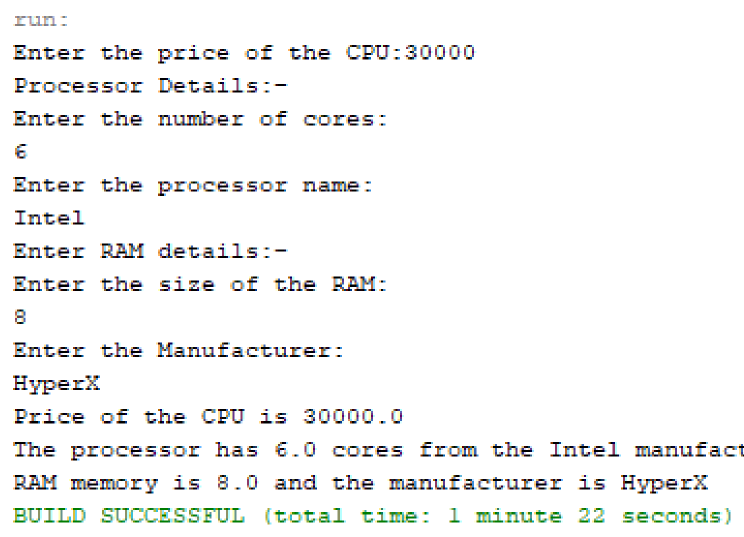
PROGRAM CODE:

CPU.java

```
import java.util.Scanner;

public class CPU {
    double price;
        public class processor
        {
            float cores;
            String mfg;
            void display(float n, String pname){
                cores = n;
                mfg = pname;
                System.out.println("The processor has "+cores+" cores from the
"+mfg+" manufacturer");
            }
        }
            static class RAM{
                float mem;
                String mfg;
                void display(float n, String pname){
                    mem = n;
```

	```
	                mfg = pname;
	                System.out.println("RAM memory is "+mem+" and the
	manufacturer is "+mfg);
	            }
	        }
	            public static void main(String[] args){
	                CPU obj = new CPU();
	                CPU.processor obj1 = obj.new processor();
	                CPU.RAM obj2 = new CPU.RAM();
	                Scanner sc = new Scanner(System.in);
	                System.out.print("Enter the price of the CPU:");
	                obj.price = sc.nextDouble();
	                System.out.println("Processor Details:-");
	                System.out.println("Enter the number of cores:");
	                float c = sc.nextFloat();
	                sc.nextLine();
	                System.out.println("Enter the processor name:");
	                String pname = sc.nextLine();

	                System.out.println("Enter RAM details:-");
	                System.out.println("Enter the size of the RAM:");
	                float mem = sc.nextFloat();
	                sc.nextLine();
	                System.out.println("Enter the Manufacturer:");
	                String mfg = sc.nextLine();

	                System.out.println("Price of the CPU is "+obj.price);
	                obj1.display(c, pname);
	                obj2.display(mem, mfg);
	            }}
	```

OUTPUT:



```
run:
Enter the price of the CPU:30000
Processor Details:-
Enter the number of cores:
6
Enter the processor name:
Intel
Enter RAM details:-
Enter the size of the RAM:
8
Enter the Manufacturer:
HyperX
Price of the CPU is 30000.0
The processor has 6.0 cores from the Intel manufacturer
RAM memory is 8.0 and the manufacturer is HyperX
BUILD SUCCESSFUL (total time: 1 minute 22 seconds)
```

COURSE OUTCOME 2
PROGRAM: 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.

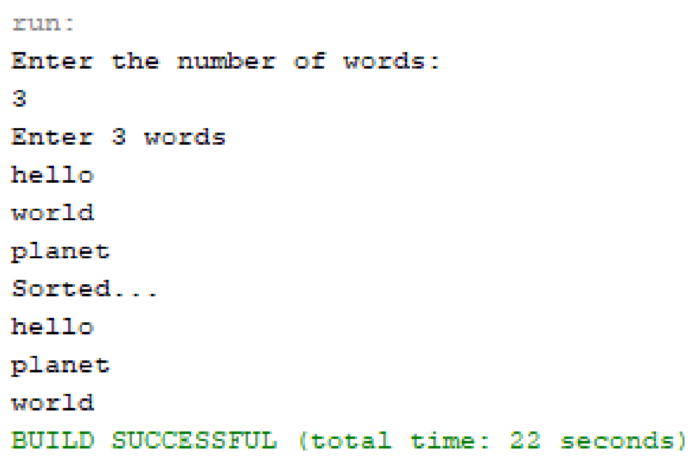
Step 6: Stop.

PROGRAM CODE:

CO2Q1.java

```
//Program to sort strings

import java.util.Scanner;


public class CO2Q1
{
    public class sort
    {
        public void display(String s[],int n)
        {
            String temp;
            String str[] = s;
            for(int i=0;i<n;i++)
                for(int j=i+1;j<n;j++)
                    if(str[i].compareTo(str[j])>0)
                    {
                        temp = str[i];
                        str[i] = str[j];
                        str[j] = temp;
                    }
            System.out.println("Sorted...");
            for(int i=0;i<n;i++)
                System.out.println(str[i]);
        }
    }
    public static void main(String[] args)
    {
        int count;
        CO2Q1 obj = new CO2Q1();
        CO2Q1.sort ob = obj.new sort();
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of words:");
        count = sc.nextInt();
        String str[] = new String[count];
        System.out.println("Enter "+count+" words");

        sc.nextLine();

        for(int i=0;i<count;i++)
        {
            str[i] = sc.nextLine();
        }

        ob.display(str,count);

    }
}
```

OUTPUT:



```
run:
Enter the number of words:
3
Enter 3 words
hello
world
planet
Sorted...
hello
planet
world
BUILD SUCCESSFUL (total time: 22 seconds)
```

PROGRAM: 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
```

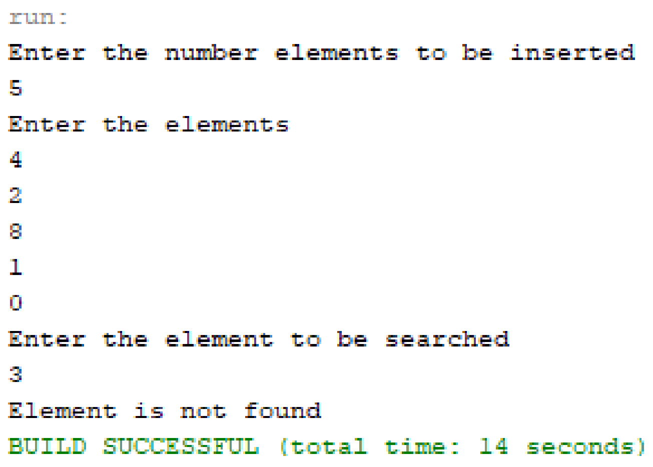
PROGRAM CODE:

CO2Q2.java

```
import java.util.Scanner;

public class CO2Q2
{
    public static void main(String[] args)
    {
        int count,flag=0;
        int pos = 0;
        String t;
        System.out.println("Enter the number elements to be inserted");
        Scanner sc = new Scanner(System.in);
        count = sc.nextInt();
        String str[] = new String[count];
        sc.nextLine();
        System.out.println("Enter the elements");
        for(int i=0;i<count;i++)
            str[i] = sc.nextLine();
        System.out.println("Enter the element to be searched");
        t = sc.nextLine();
        for(int i=0;i<count;i++)
        {
            if(str[i].equals(t))
            {
                flag = 1;
                pos = i;
                break;
            }
        }
```

```
          if(flag==1)
          {
             System.out.println("Element is in position "+(pos+1));
          }
          else
          {
             System.out.println("Element is not found");
          }
      }
  }
```

OUTPUT:



```
run:
Enter the number elements to be inserted
5
Enter the elements
4
2
8
1
0
Enter the element to be searched
3
Element is not found
BUILD SUCCESSFUL (total time: 14 seconds)
```

PROGRAM: 8

AIM: Perform string manipulations

ALGORITHM :

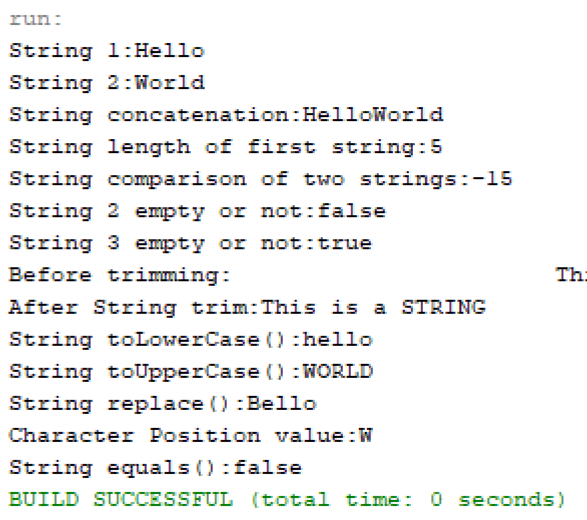
```
Step 1: Start
Step 2: Take the strings provided by the user and concatenate them.
Step 4: Display the length of the first string.
Step 5: Display the comparison of two strings.
Step 6: Display the strings were they are empty or not.
Step 7: Display the trimmed string form.
Step 8: Display the string with lower case.
Step 9: Display the string with upper case.
Step 10: Display the string after replacing all the ‘H’ characters with ‘B’ character.
Step 11: Stop
```

PROGRAM CODE:

CO2Q3.java

```java
//String Manipulation

public class CO2Q3
{
    public static void main(String[] args)
    {
        String s1 = "Hello";
        String s2 = "World";
        String s3 = "";
        String s4 = "                     This is a STRING        ";
        System.out.println("String 1:"+s1);
        System.out.println("String 2:"+s2);
        System.out.println("String concatenation:"+s1.concat(s2));
        System.out.println("String length of first string:"+s1.length());
        System.out.println("String comparison of two 
strings:"+s1.compareTo(s2));
        System.out.println("String 2 empty or not:"+s2.isEmpty());
        System.out.println("String 3 empty or not:"+s3.isEmpty());
        System.out.println("Before trimming:"+s4);
        System.out.println("After String trim:"+s4.trim());
        System.out.println("String toLowerCase():"+s1.toLowerCase());
        System.out.println("String toUpperCase():"+s2.toUpperCase());
        System.out.println("String replace():"+(s1.replace("H", "B")));
        System.out.println("Character Position value:"+s2.charAt(0));
        System.out.println("String equals():"+s1.equals(s2));

    }
}
```

OUTPUT:



```
run:
String 1:Hello
String 2:World
String concatenation:HelloWorld
String length of first string:5
String comparison of two strings:-15
String 2 empty or not:false
String 3 empty or not:true
Before trimming:                        This is a STRING
After String trim:This is a STRING
String toLowerCase():hello
String toUpperCase():WORLD
String replace():Bello
Character Position value:W
String equals():false
BUILD SUCCESSFUL (total time: 0 seconds)
```

PROGRAM: 9

AIM: Program to create a class for Employee having attributes Empid, Name, Salary, Address. Read n employee information and Search for an employee given Empid, using the concept of Array of Objects.

ALGORITHM :

Step 1: Start

Step 2: Search the ‘Empid’ attribute of the list of Employee Objects for the ‘Empid’ provided by the user.

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

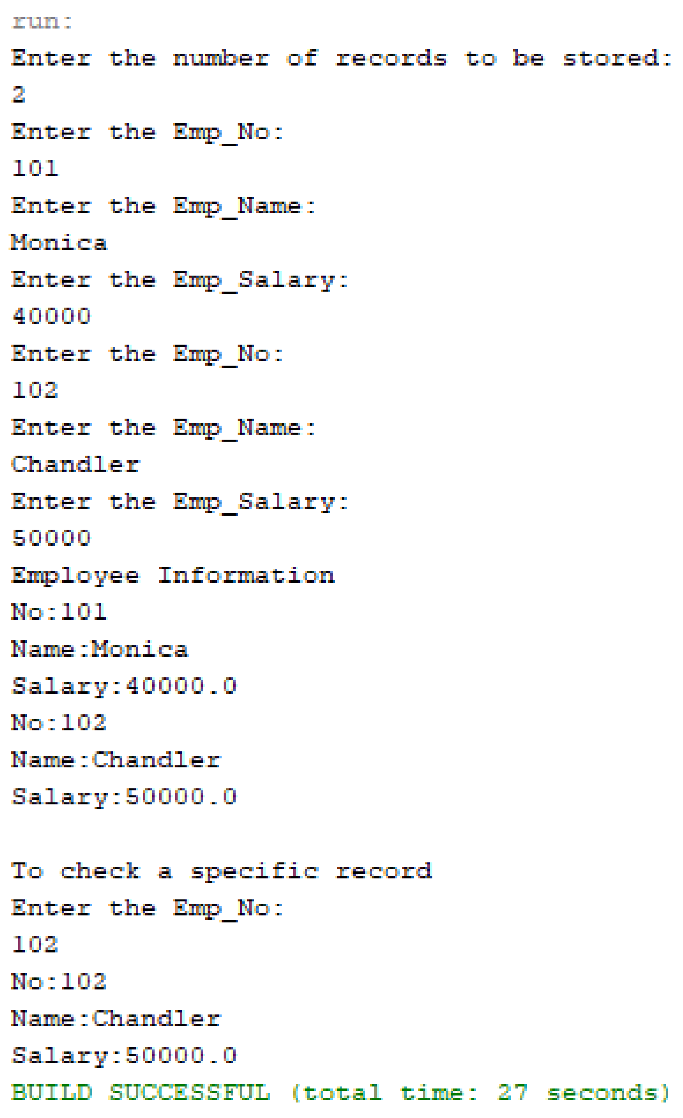
Step 4: Stop

PROGRAM CODE:

CO2Q4.java

```
//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.
import java.util.*;

public class CO2Q4{

                int[] eNo = new int[20];
                int count,i,e;
                String[] eName = new String[50];
                float[] eSalary = new float[20];


                void getinfo(int c){
                    Scanner s = new Scanner(System.in);
                    count=c;
                      for(i=0;i<c;i++){
                          System.out.println("Enter the Emp_No:");
                          eNo[i]=s.nextInt();
                          System.out.println("Enter the Emp_Name:");
                          eName[i]=s.next();
                          System.out.println("Enter the Emp_Salary:");
                    eSalary[i]=s.nextFloat();
                }
            }

            void printinfo(int c){
                count =c;
                System.out.println("Employee Information");
                for(i=0;i<count;i++)
                {
                  System.out.println("No:"+eNo[i]);
                  System.out.println("Name:"+eName[i]);
                  System.out.println("Salary:"+eSalary[i]);
                }
            }

            void displayinfo(int emp_no, int c) {
                    int flag=0;
                    e = emp_no;
                    count = c;
                    for(i=0;i<count;i++)
                    {
                            if(eNo[i]==e)
                            {
                            System.out.println("No:"+eNo[i]);
                        System.out.println("Name:"+eName[i]);
                        System.out.println("Salary:"+eSalary[i]);
                        flag++;
                            }

                    }
                    if(flag==0)
                            System.out.println("Record Not Found!");
           }

            public static void main(String[] args){
                CO2Q4 obj = new CO2Q4();
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the number of records to be
stored:");
                obj.count = sc.nextInt();
                obj.getinfo(obj.count);
                obj.printinfo(obj.count);
                System.out.println("\nTo check a specific record");
                System.out.println("Enter the Emp_No:");
                int e = sc.nextInt();
                obj.displayinfo(e,obj.count);
                sc.close();
            }
        }
```

OUTPUT:



```
run:
Enter the number of records to be stored:
2
Enter the Emp_No:
101
Enter the Emp_Name:
Monica
Enter the Emp_Salary:
40000
Enter the Emp_No:
102
Enter the Emp_Name:
Chandler
Enter the Emp_Salary:
50000
Employee Information
No:101
Name:Monica
Salary:40000.0
No:102
Name:Chandler
Salary:50000.0

To check a specific record
Enter the Emp_No:
102
No:102
Name:Chandler
Salary:50000.0
BUILD SUCCESSFUL (total time: 27 seconds)
```

PROGRAM NO: 10

AIM:

To find 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

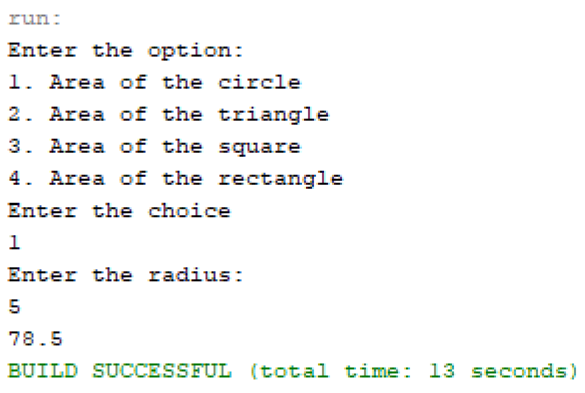
Step 4: Display the areas of each shapes.

PROGRAM CODE:

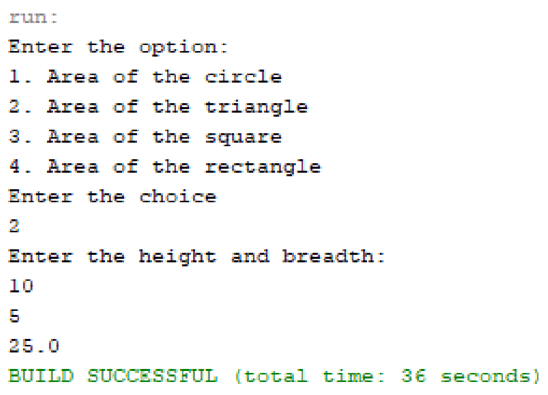
CO3Q1.java

```
import java.util.Scanner;

public class CO3Q1
{
    double area(float r)
    {
        double pi = 3.14;
        double ar;
        ar = pi*r*r;
        return ar;
    }
    double area(float h,float b)
    {
        double ar;
        ar = (h*b)/2;
        return ar;
    }
    double area(double s)
    {
        double ar;
        ar = s*s;
        return ar;
    }
    double area(double l,double br)
    {
        double ar;
        ar = l*br;
        return ar;
    }
    public static void main(String[] args)
    {
        CO3Q1 obj = new CO3Q1();
        int ch;
        float r,h,b;
        double s,l,br;
        System.out.println("Enter the option:");
        System.out.println("1. Area of the circle");
        System.out.println("2. Area of the triangle");
        System.out.println("3. Area of the square");
        System.out.println("4. Area of the rectangle");
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the choice");
        ch = sc.nextInt();
        switch(ch)
        {
           case 1: System.out.println("Enter the radius:");
                 r = sc.nextFloat();
                 System.out.println(obj.area(r));
                 break;
           case 2: System.out.println("Enter the height and breadth:");
                 h = sc.nextFloat();
                 b = sc.nextFloat();
                 System.out.println(obj.area(h,b));
                 break;
           case 3: System.out.println("Enter the length of the side:");
                 s = sc.nextDouble();
                 System.out.println(obj.area(s));
                 break;
           case 4: System.out.println("Enter the length and breadth:");
                 l = sc.nextDouble();
                 br = sc.nextDouble();
                 System.out.println(obj.area(l,br));
                 break;
           default: System.out.println("Invalid choice.");
        }
    }
}
```

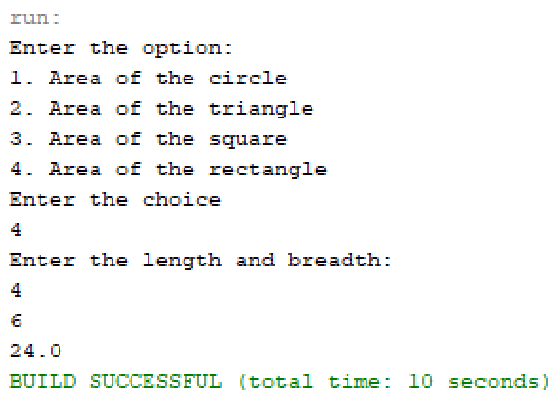
OUTPUT:



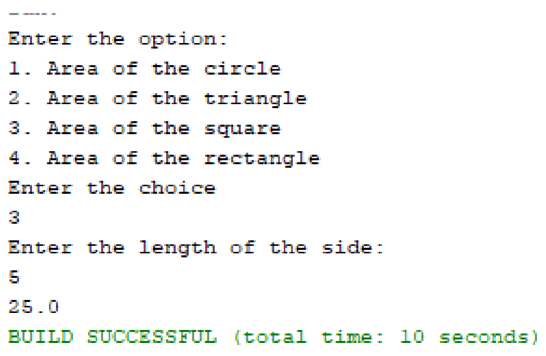
```
run:
Enter the option:
1. Area of the circle
2. Area of the triangle
3. Area of the square
4. Area of the rectangle
Enter the choice
1
Enter the radius:
5
78.5
BUILD SUCCESSFUL (total time: 13 seconds)
```



```
run:
Enter the option:
1. Area of the circle
2. Area of the triangle
3. Area of the square
4. Area of the rectangle
Enter the choice
2
Enter the height and breadth:
10
5
25.0
BUILD SUCCESSFUL (total time: 36 seconds)
```



```
run:
Enter the option:
1. Area of the circle
2. Area of the triangle
3. Area of the square
4. Area of the rectangle
Enter the choice
4
Enter the length and breadth:
4
6
24.0
BUILD SUCCESSFUL (total time: 10 seconds)
```



```
Enter the option:
1. Area of the circle
2. Area of the triangle
3. Area of the square
4. Area of the rectangle
Enter the choice
3
Enter the length of the side:
5
25.0
BUILD SUCCESSFUL (total time: 10 seconds)
```

RESULT:

The program is successfully executed and the output is verified.

PROGRAM NO: 11

AIM:

To 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 the program.

Step.2: Define a class '*Employee*' with data members Empid, Name, Salary, Address and a constructor to initialize these members.

Step.3: Define a 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 a method Display() to display all the data members.

Step.4: Define a main() to create array of objects for the class to display the details of 'N' teachers.

Step.5: Stop the program.

PROGRAM CODE:

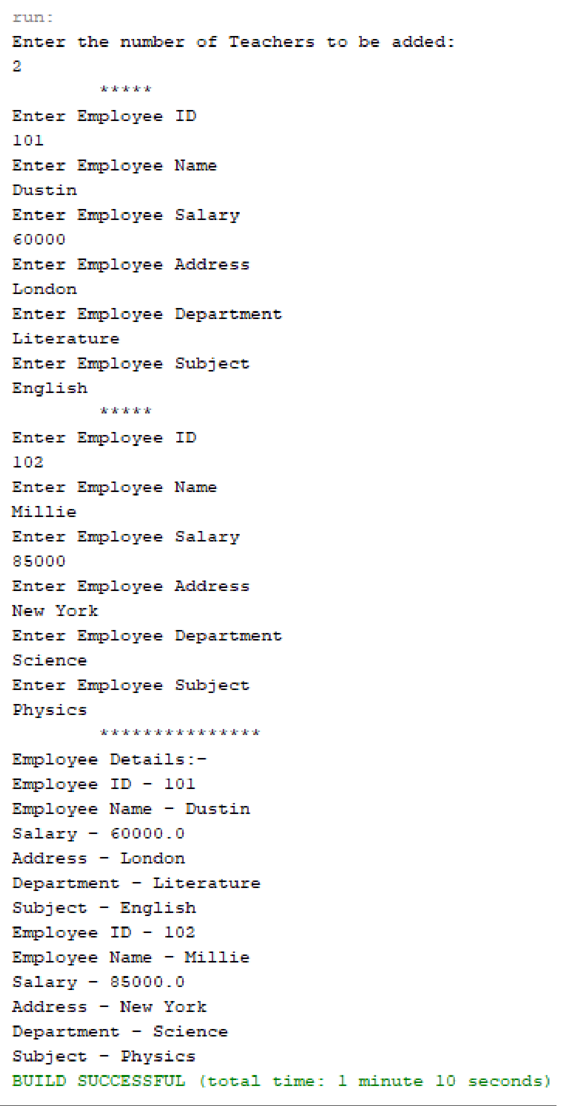
employee.java

```
class Employee2
{
    int Empid;
    String Name;
    float Salary;
    String Address;

    Employee2()
    {

    }
    public Employee2(int id, String name, float sal, String addr)
    {
        Empid = id;
        Name = name;
        Salary = sal;
        Address = addr;
    }
```

```
}
class Teacher extends Employee2
{
    String department;
    String Subjects;

    public Teacher(int id, String name, float sal, String addr, String dept, String sub)
    {
        super(id, name, sal, addr);
        department = dept;
        Subjects = sub;

    }

    Teacher()
    {

    }

    public void display()
    {
        System.out.println("Employee ID - "+Empid);
        System.out.println("Employee Name - "+Name);
        System.out.println("Salary - "+Salary);
        System.out.println("Address - "+Address);
        System.out.println("Department - "+department);
        System.out.println("Subject - "+Subjects);
    }

}
public class CO3Q2
{
    public static void main(String[] args)
    {
        int n;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of Teachers to be added:");
        n = sc.nextInt();
        Teacher obj[] = new Teacher[n];
        for (int i=0; i<n; i++)
        {
            obj[i] = new Teacher();
        }
        for(int i=0;i<n;i++)
        {
            System.out.println("\t*****");
            System.out.println("Enter Employee ID");
            obj[i].Empid = sc.nextInt();
            sc.nextLine();
            System.out.println("Enter Employee Name");
            obj[i].Name = sc.nextLine();
            System.out.println("Enter Employee Salary");
            obj[i].Salary = sc.nextFloat();
            sc.nextLine();
```

```
                System.out.println("Enter Employee Address");
                obj[i].Address = sc.nextLine();
                System.out.println("Enter Employee Department");
                obj[i].department = sc.nextLine();
                System.out.println("Enter Employee Subject");
                obj[i].Subjects = sc.nextLine();
            }
            System.out.println("\t***************");
            System.out.println("Employee Details:-");
            for(int i=0;i<n;i++)
                obj[i].display();
        }
    }
```

OUTPUT:



```
run:
Enter the number of Teachers to be added:
2
        *****
Enter Employee ID
101
Enter Employee Name
Dustin
Enter Employee Salary
60000
Enter Employee Address
London
Enter Employee Department
Literature
Enter Employee Subject
English
        *****
Enter Employee ID
102
Enter Employee Name
Millie
Enter Employee Salary
85000
Enter Employee Address
New York
Enter Employee Department
Science
Enter Employee Subject
Physics
        ***************
Employee Details:-
Employee ID - 101
Employee Name - Dustin
Salary - 60000.0
Address - London
Department - Literature
Subject - English
Employee ID - 102
Employee Name - Millie
Salary - 85000.0
Address - New York
Department - Science
Subject - Physics
BUILD SUCCESSFUL (total time: 1 minute 10 seconds)
```

RESULT:

The program is successfully executed and the output is verified.

PROGRAM NO: 12

AIM:

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

ALGORITHM:

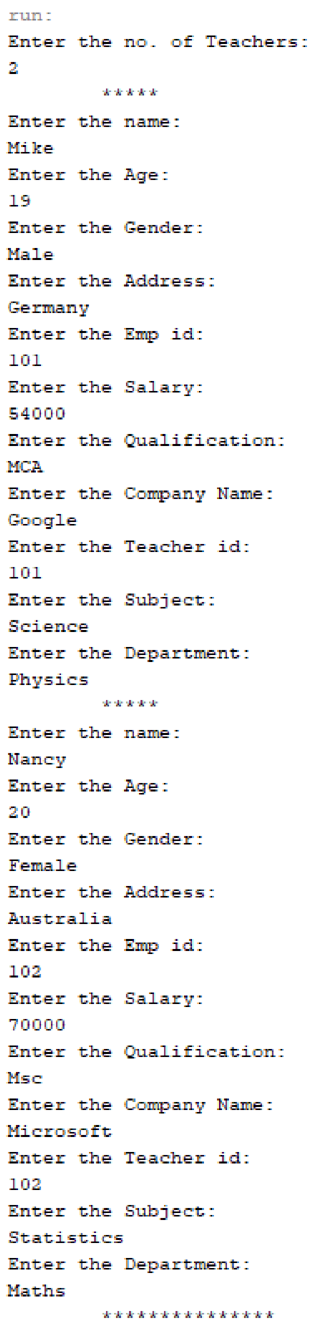
```
Step 1: Start
Step 2: Create a class named ‘Person’ with data members name, gender, address and age
& a constructor to initialize them.
Step 3: Create a class named ‘Employee’ which is derived from Person, with data members
empid, cmpnyname, qualification and sal & a constructor Employee() to initialize 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:

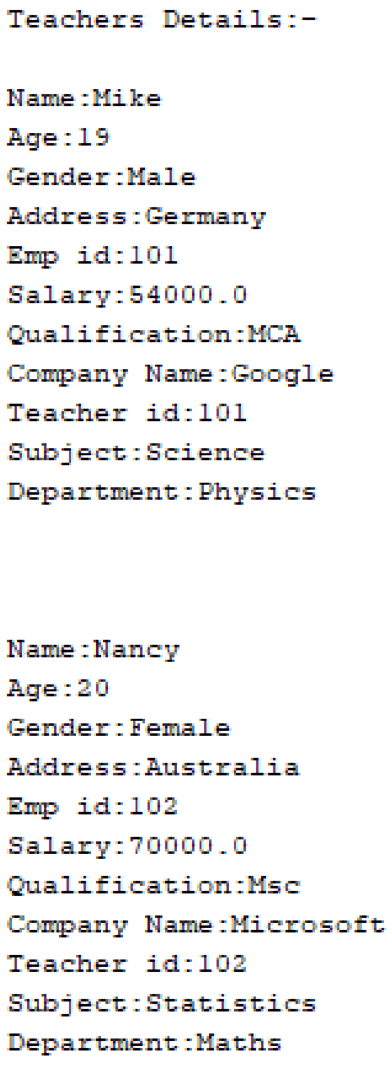
CO3Q3.java	
	```
	import java.util.Scanner;

	class Person
	{
	    String Name;
	    String Gender;
	    String Address;
	    int Age;
	    Person()
	    {
	}
	```

```
   Person(String name, String gender, String addr, int age){
      Name = name;
      Gender = gender;
      Address = addr;
      Age = age;
   }}
class Employee extends Person
{
   int Empid;
   String Company_name;
   String Qualification;
   float Salary;
   Employee()
   {    }
   public Employee(String name, String gender, String addr, int age) {
      super(name, gender, addr, age);
   }
   public Employee(int id,String name, String qual, float sal){
      Empid = id;
      Company_name = name;
      Qualification = qual;
      Salary = sal;
   }}
class Teacher extends Employee{
   String Subject;
   String Department;
   String Teachersid;
   Teacher()
   {    }
   Teacher(String sub, String dept, String id){
      Subject = sub;
      Department = dept;
      Teachersid = id;
   }
   public void display(){
      System.out.println("Name:" + Name);
         System.out.println("Age:" + Age);
         System.out.println("Gender:" + Gender);
         System.out.println("Address:" + Address);
         System.out.println("Emp id:" + Empid);
         System.out.println("Salary:" + Salary);
         System.out.println("Qualification:" + Qualification);
         System.out.println("Company Name:" + Company_name);
         System.out.println("Teacher id:" + Teachersid);
         System.out.println("Subject:" + Subject);
         System.out.println("Department:" + Department);
      System.out.println("\n\n");
   }}
public class CO3Q3 {
   public static void main(String[] args) {
    int n;
    System.out.println("Enter the no. of Teachers:");
    Scanner sc = new Scanner(System.in);
    n = sc.nextInt();
    Teacher obj[] = new Teacher[n];
    for(int i=0;i<n;i++)
        obj[i] = new Teacher();
    sc.nextLine();
    for(int i=0;i<n;i++){
        System.out.println("\t*****");
        System.out.println("Enter the name:");
        obj[i].Name = sc.nextLine();
        System.out.println("Enter the Age:");
        obj[i].Age = sc.nextInt();
        sc.nextLine();
        System.out.println("Enter the Gender:");
        obj[i].Gender = sc.nextLine();
        System.out.println("Enter the Address:");
        obj[i].Address = sc.nextLine();
        System.out.println("Enter the Emp id:");
        obj[i].Empid = sc.nextInt();
        System.out.println("Enter the Salary:");
        obj[i].Salary = sc.nextFloat();
        sc.nextLine();
        System.out.println("Enter the Qualification:");
        obj[i].Qualification = sc.nextLine();
        System.out.println("Enter the Company Name:");
        obj[i].Company_name = sc.nextLine();
        System.out.println("Enter the Teacher id:");
        obj[i].Teachersid = sc.nextLine();
        System.out.println("Enter the Subject:");
        obj[i].Subject = sc.nextLine();
        System.out.println("Enter the Department:");
        obj[i].Department = sc.nextLine();
    }
    System.out.println("\t***************");
    System.out.println("Teachers Details:-\n");
    for(int i=0;i<n;i++)
        obj[i].display();
}}
```

OUTPUT:



```
run:
Enter the no. of Teachers:
2
        *****
Enter the name:
Mike
Enter the Age:
19
Enter the Gender:
Male
Enter the Address:
Germany
Enter the Emp id:
101
Enter the Salary:
54000
Enter the Qualification:
MCA
Enter the Company Name:
Google
Enter the Teacher id:
101
Enter the Subject:
Science
Enter the Department:
Physics
        *****
Enter the name:
Nancy
Enter the Age:
20
Enter the Gender:
Female
Enter the Address:
Australia
Enter the Emp id:
102
Enter the Salary:
70000
Enter the Qualification:
Msc
Enter the Company Name:
Microsoft
Enter the Teacher id:
102
Enter the Subject:
Statistics
Enter the Department:
Maths
        ***************
```



```
Teachers Details:-

Name:Mike
Age:19
Gender:Male
Address:Germany
Emp id:101
Salary:54000.0
Qualification:MCA
Company Name:Google
Teacher id:101
Subject:Science
Department:Physics



Name:Nancy
Age:20
Gender:Female
Address:Australia
Emp id:102
Salary:70000.0
Qualification:Msc
Company Name:Microsoft
Teacher id:102
Subject:Statistics
Department:Maths
```

RESULT:

The program is successfully executed and the output is verified.

PROGRAM NO: 13

AIM:

To 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 literature create an object of literature
type and object of type fiction if fiction is chosen.
Step 7 : Stop
```

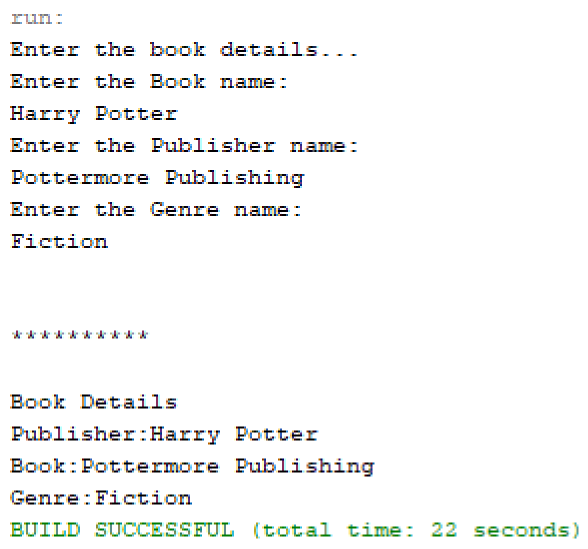
PROGRAM CODE:

CO3Q4.java	
	```
	import java.util.Scanner;


	public class CO3Q4
	{
	    public class publisher
	    {
	        String pub_name;
	        publisher(){}
	        publisher(String name)
	        {
	            pub_name = name;
	        }
	    }
	   static public class Book extends publisher
	    {
	        String book_name;
	        Book(){}
	        Book(String bname, String pname)
	```

```
        {
            super(pname);
            book_name = bname;
        }
        public void display()
        {
            System.out.println("\n\n**********\n\nBook Details");
            System.out.println("Publisher:"+ pub_name);
            System.out.println("Book:"+book_name);
        }
    }
    static public class literature extends Book
    {
        String book_genre;

        public literature() {}
        literature(String name,String book,String genre)
        {
            super(name,book);
            book_genre = genre;
            super.display();
            System.out.println("Genre:"+book_genre);
        }

    }
    static public class fiction extends Book
    {
        String book_genre;
        fiction(){}
        fiction(String name,String book,String genre)
        {
            super(name,book);
            book_genre = genre;
            super.display();
            System.out.println("Genre:"+book_genre);
        }
    }


    public static void main(String[] args)
    {
        String name,book,genre;

        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the book details...");
        System.out.println("Enter the Book name:");
        book = sc.nextLine();
        System.out.println("Enter the Publisher name:");
        name = sc.nextLine();
        System.out.println("Enter the Genre name:");
```

```
          genre = sc.nextLine();
          fiction obj;
          literature ob;
          if(genre.toLowerCase().equals("fiction"))
          {
              obj = new fiction(name,book,genre);
          }
          else if(genre.toLowerCase().equals("literature"))
              ob = new literature(name,book,genre);
          else
              System.out.println("Enter Fiction or Literature");
      }
  }
```

OUTPUT:



```
run:
Enter the book details...
Enter the Book name:
Harry Potter
Enter the Publisher name:
Pottermore Publishing
Enter the Genre name:
Fiction


**********

Book Details
Publisher:Harry Potter
Book:Pottermore Publishing
Genre:Fiction
BUILD SUCCESSFUL (total time: 22 seconds)
```

RESULT:

The program is successfully executed and the output is verified.

PROGRAM NO: 14

AIM:

To 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 the program.

Step.2: Define a class ‘*Student*’ which will read a student’s academic information from the user.

Step.3: Define another class ‘*Sports*’ that extends ‘*Student*’ and reads the sports data of the student.

Step.4: Define another interface ‘R*esults*’ that extends ‘*Sports*’and has a *Display()* to display the profile, academic score and sports score of the student.

Step.5: Define a main () method to create objects for the above classes and to call the associated member methods.

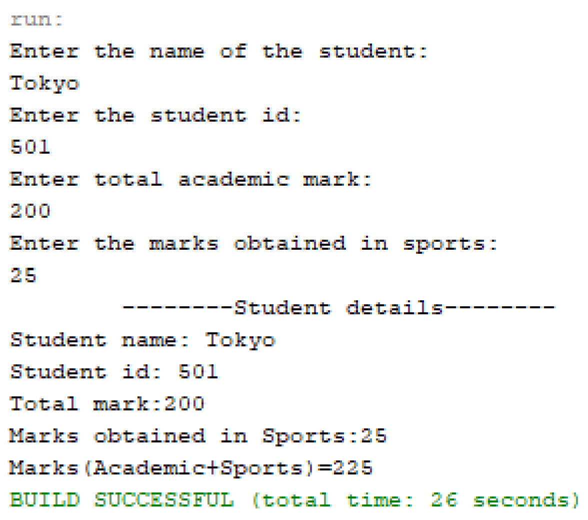
Step.6: Stop the program.

PROGRAM CODE:

CO3Q5.java

```java
import java.util.Scanner;

interface results{
    void getdata();
    int display();
}
class Student implements results{
    int std_id,std_tmark;
    String std_name;
    public void getdata(){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the name of the student:");
        std_name = sc.nextLine();
        System.out.println("Enter the student id:");
        std_id = sc.nextInt();
        System.out.println("Enter total academic mark:");
        std_tmark = sc.nextInt();
    }
    public int display(){
        System.out.println("\t--------Student details--------");
        System.out.println("Student name: " +std_name);
        System.out.println("Student id: " +std_id);
        System.out.println("Total mark:" +std_tmark);
        return std_tmark;
    }}
class Sports implements results{
    int tmarks;
    public void getdata() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the marks obtained in sports:");
        tmarks = sc.nextInt();
    }
    public int display(){
        System.out.println("Marks obtained in Sports:"+ tmarks);
        return tmarks;
    }}
public class CO3Q5 {
    public static void main(String[] args) {
        int mark;
        Student ob = new Student();
        Sports obj = new Sports();
        ob.getdata();
        obj.getdata();
        mark = ob.display();
        mark = mark + obj.display();
        System.out.println("Marks(Academic+Sports)="+ mark);
    }}
```

OUTPUT:



```
run:
Enter the name of the student:
Tokyo
Enter the student id:
501
Enter total academic mark:
200
Enter the marks obtained in sports:
25
        --------Student details--------
Student name: Tokyo
Student id: 501
Total mark:200
Marks obtained in Sports:25
Marks(Academic+Sports)=225
BUILD SUCCESSFUL (total time: 26 seconds)
```

RESULT:

The program is successfully executed and the output is verified.

PROGRAM NO: 15

AIM:

To 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 the program.

Step.2: Define an interface '*prop*' with methods to read inputs and calculate area and perimeter.

Step.3: Define a class '*Rectangle*' that extends *Circle* to initialize its data members l, b and to calculate and display the area and perimeter of a rectangle.

Step.4: Define a main () to create objects for the above classes to invoke its member methods to print the results.

Step.5: Stop the program.

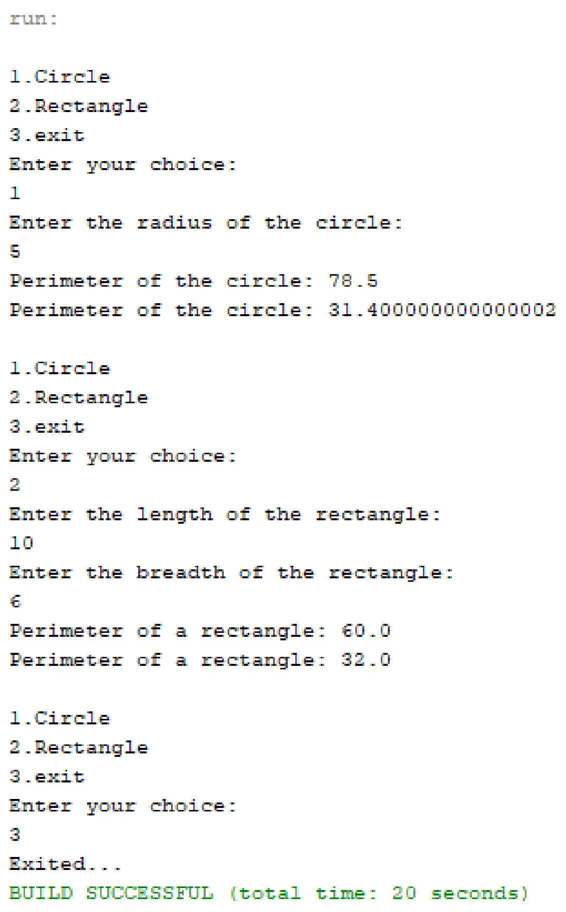
PROGRAM CODE:

CO3Q6.java

```
import java.util.Scanner;
interface prop{
    void getdata();
    void area();
    void perimeter();
}
class Circle implements prop{
    double pi = 3.14;
    double r;
    Scanner sc = new Scanner(System.in);
    @Override
    public void getdata(){
        System.out.println("Enter the radius of the circle:");
        r = sc.nextDouble();
    }
    @Override
    public void perimeter(){
        System.out.println("Perimeter of the circle: "+(2*pi*r));
    }
    @Override
    public void area(){
        System.out.println("Perimeter of the circle: "+(pi*r*r));
    }
}
```

```
class Rectangle implements prop{
    double l,b;
    Scanner sc = new Scanner(System.in);
    @Override
    public void getdata(){
        System.out.println("Enter the length of the rectangle:");
        l = sc.nextDouble();
        System.out.println("Enter the breadth of the rectangle:");
        b = sc.nextDouble();
    }
    @Override
    public void area(){
        System.out.println("Perimeter of a rectangle: "+(l*b));
    }
    @Override
    public void perimeter(){
        System.out.println("Perimeter of a rectangle: "+(2*(l+b)));
    }
}
public class CO3Q6 {
    public static void main(String[] args) {
        int ch;
        Scanner sc = new Scanner(System.in);
        Circle ob = new Circle();
        Rectangle obj = new Rectangle();
        do{
            System.out.println("\n1.Circle\n2.Rectangle\n3.exit");
            System.out.println("Enter your choice:");
            ch = sc.nextInt();
            switch(ch){
                case 1 :ob.getdata();
                        ob.area();
                        ob.perimeter();
                        break;
                case 2 :obj.getdata();
                        obj.area();
                        obj.perimeter();
                        break;
                case 3 :System.out.println("Exited...");
                        System.exit(0);
            }
        }while(true);
    }
}
```

OUTPUT:



```
run:

1.Circle
2.Rectangle
3.exit
Enter your choice:
1
Enter the radius of the circle:
5
Perimeter of the circle: 78.5
Perimeter of the circle: 31.400000000000002

1.Circle
2.Rectangle
3.exit
Enter your choice:
2
Enter the length of the rectangle:
10
Enter the breadth of the rectangle:
6
Perimeter of a rectangle: 60.0
Perimeter of a rectangle: 32.0

1.Circle
2.Rectangle
3.exit
Enter your choice:
3
Exited...
BUILD SUCCESSFUL (total time: 20 seconds)
```

RESULT:

The program is successfully executed and the output is verified.

PROGRAM NO: 16

AIM:

To 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 the program.

Step.2: Define an interface *calc* with a method calculate().

Step.3: Define a class *bill* that implements *calc* to calculate the total amount for each product and has methods to generate a bill as given in the question.

Step.4: Define a main () to create objects for the class.

Step.5: Invoke the above methods by passing the data collected from user to generate the required bill.

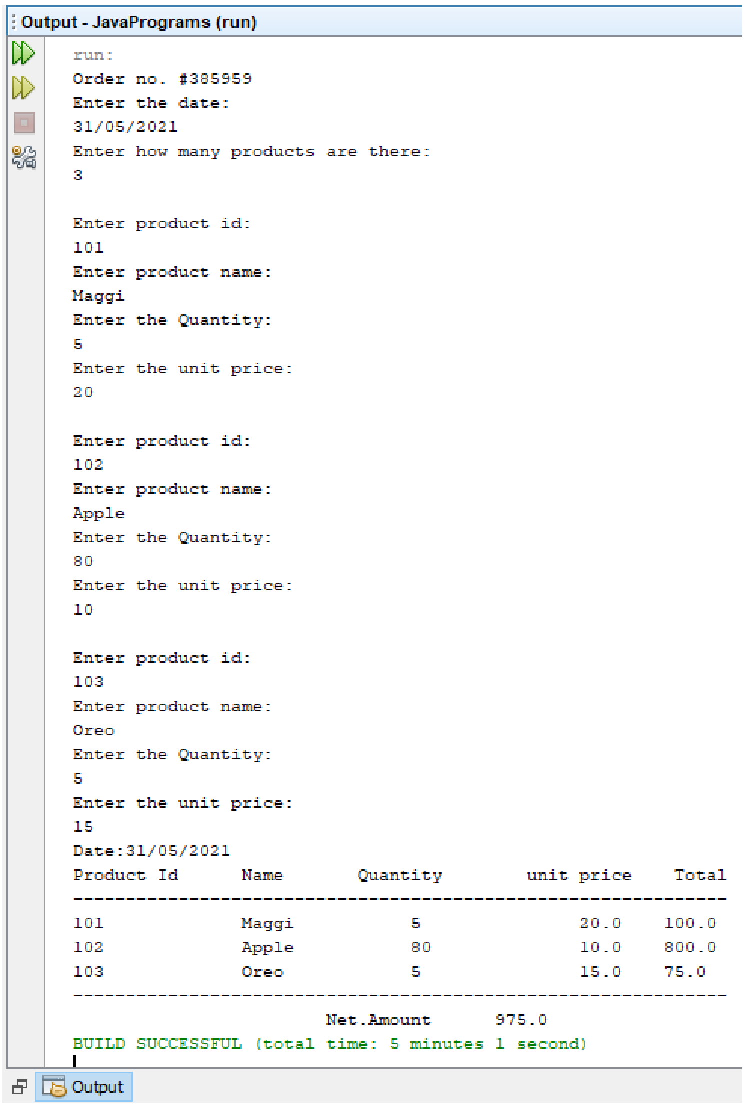
Step.6: Stop the program.

PROGRAM CODE:

CO3Q7.java

```
import java.util.Scanner;
interface calc{
    void calculate();
}
class bill implements calc{
    String date,name,p_id;
    int quantity;
    double unit_price,total,namount=0;
    Scanner sc = new Scanner(System.in);
    public void getdata(){
        System.out.println("\nEnter product id:");
        p_id = sc.nextLine();
        System.out.println("Enter product name:");
        name = sc.nextLine();
        System.out.println("Enter the Quantity:");
        quantity = sc.nextInt();
        System.out.println("Enter the unit price:");
        unit_price = sc.nextDouble();
    }
    @Override
    public void calculate(){
        total = quantity * unit_price;
    }
    public void display(){
        System.out.println(p_id+"\t\t"+name+"\t\t"+quantity+"\t\t"+unit_price+"\t"+total);
    }
}
public class CO3Q7 {
    public static void main(String[] args) {
        int n,i;
        double namount=0,t;
        int ran;
        String date;
        t = Math.random() *1000000;
        ran = (int) t;
        Scanner sc = new Scanner(System.in);
        System.out.println("Order no. #"+ran);
        System.out.println("Enter the date:");
        date = sc.nextLine();
        System.out.println("Enter how many products are there:");
        n = sc.nextInt();
        bill ob[] = new bill[n];
        for(i=0;i<n;i++)
            ob[i] = new bill();
        for(i=0;i<n;i++){
            ob[i].getdata();
            ob[i].calculate();
        }
        System.out.println("Date:"+date);
        System.out.println("Product Id \tName\t   Quantity\t   unit price\t Total ");
        System.out.println("-------------------------------------------------------------");
        for(i=0;i<n;i++){
            ob[i].display();
            namount += ob[i].total;
        }
        System.out.println("-------------------------------------------------------------");
        System.out.println("\t\t\tNet.Amount\t"+ namount);

    }}
```

OUTPUT:



Output - JavaPrograms (run)

```
run:
Order no. #385959
Enter the date:
31/05/2021
Enter how many products are there:
3

Enter product id:
101
Enter product name:
Maggi
Enter the Quantity:
5
Enter the unit price:
20

Enter product id:
102
Enter product name:
Apple
Enter the Quantity:
80
Enter the unit price:
10

Enter product id:
103
Enter product name:
Oreo
Enter the Quantity:
5
Enter the unit price:
15
Date:31/05/2021
Product Id      Name        Quantity         unit price     Total
-------------------------------------------------------------------
101             Maggi           5                 20.0     100.0
102             Apple           80                10.0     800.0
103             Oreo            5                 15.0     75.0
-------------------------------------------------------------------
                         Net.Amount      975.0
BUILD SUCCESSFUL (total time: 5 minutes 1 second)
```

RESULT:

The program is successfully executed and the output is verified.

PROGRAM NO-17

AIM:

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

ALGORITHM:

Step 1: Start.

Step 2: Define a 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.

CODE:

Graphics.Shapes.java	
	```
	package Graphics;
	interface figures{
	    void Rectangle(double a,double b);
	    void Triangle(double a,double b);
	    void Square(double a);
	    void Circle(double a);
	}
	public class Shapes implements figures{
	    @Override
	    public void Rectangle(double a,double b){
	        System.out.println("Area:"+(a*b));
	    }
	    @Override
	    public void Triangle(double a,double b){
	        System.out.println("Area:"+((a*b)/2));
	    }
	    @Override
	    public void Square(double a) {
	        System.out.println("Area:"+(a*a));
	    }
	    @Override
	    public void Circle(double a) {
	        double pi = 3.14;
	        System.out.println("Area:"+(pi*a*a));
	    }
	```

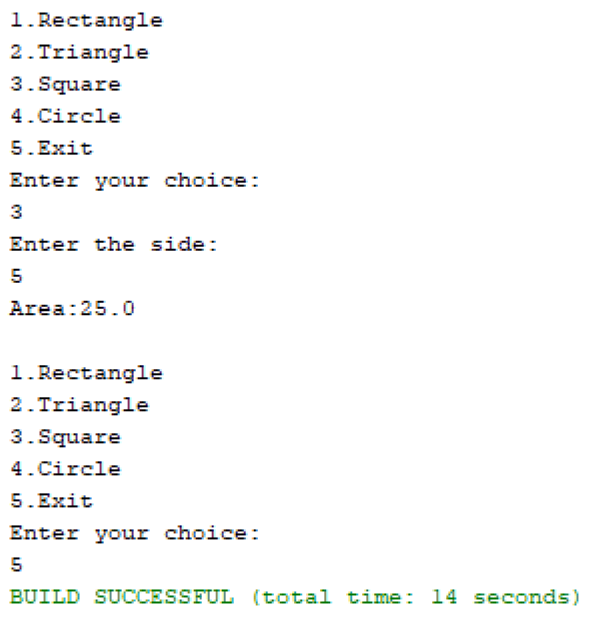
	```
	    public static void main(String[] args) {
	    }
	}
	```

CO4Q1.java	
	```
	import Graphics.Shapes;
	import java.util.Scanner;

	public class CO4Q1 {
	    public static void main(String[] args) {
	        int ch;
	        double l,b;
	        Shapes ob = new Shapes();
	        Scanner sc = new Scanner(System.in);
	        do
	        {
	System.out.println("\n1.Rectangle\n2.Triangle\n3.Square\n4.Circle\n5.
	Exit\nEnter your choice:");
	            ch = sc.nextInt();
	            switch(ch){
	                case 1:System.out.println("Enter the length and breadth:");
	                        l = sc.nextDouble();
	                        b = sc.nextDouble();
	                        ob.Rectangle(l, b);
	                        break;
	                case 2:System.out.println("Enter the breadth and height:");
	                        l = sc.nextDouble();
	                        b = sc.nextDouble();
	                        ob.Triangle(l, b);
	                        break;
	                case 3:System.out.println("Enter the side:");
	                        l = sc.nextDouble();
	                        ob.Square(l);
	                        break;
	                case 4:System.out.println("Enter the radius:");
	                        l = sc.nextDouble();
	                        ob.Circle(l);
	                        break;
	                case 5:System.exit(0);
	                        break;
	                default:System.out.println("Invalid choice");
	            }
	        }while(true);
	    }
	}
	```

RESULT:

The above program is successfully executed and the output is obtained.

OUTPUT:



```
1.Rectangle
2.Triangle
3.Square
4.Circle
5.Exit
Enter your choice:
3
Enter the side:
5
Area:25.0

1.Rectangle
2.Triangle
3.Square
4.Circle
5.Exit
Enter your choice:
5
BUILD SUCCESSFUL (total time: 14 seconds)
```

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 access the above module and print the output.

Step 5:Stop

CODE:

Arithmetic.operations.java	package arithmetic;
	interface basic{
	    void addition(double a,double b);
	    void subtraction(double a,double b);
	    void multiplication(double a,double b);
	    void division(double a,double b);
	}
	public class operations implements basic{
	    public void addition(double a, double b) {
	        System.out.println(a+" + "+b+" = "+(a+b));

	` }`
	` public void subtraction(double a, double b) {`
	`     System.out.println(a+" - "+b+" = "+(a-b));`
	` }`
	` public void multiplication(double a, double b) {`
	`     System.out.println(a+" x "+b+" = "+(a*b));`
	` }`
	` public void division(double a, double b) {`
	`     System.out.println(a+" / "+b+" = "+(a/b));`
	` }}`

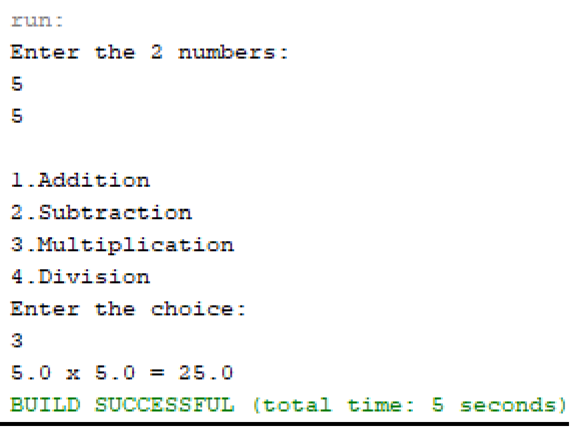
CO4Q2.java	`package c04q2;`
	`//Create an Arithmetic package that has classes and interfaces for the 4 basic arithmetic`
	`//operations. Test the package by implementing all operations on two given numbers`
	`import arithmetic.operations;`
	`import java.util.Scanner;`
	`public class C04Q2 {`
	`    public static void main(String[] args) {`
	`        double n1,n2;`
	`        int ch;`
	`        operations ob = new operations();`
	`        Scanner sc = new Scanner(System.in);`
	`        System.out.println("Enter the 2 numbers:");`
	`        n1 = sc.nextDouble();`
	`        n2 = sc.nextDouble();`
	`System.out.println("\n1.Addition\n2.Subtraction\n3.Multiplication\n4.Division\nEnter the choice:");`
	`        ch = sc.nextInt();`
	`        switch(ch){`
	`            case 1:ob.addition(n1,n2);`
	`                    break;`

	case 2:ob.subtraction(n1,n2);
	        break;
	case 3:ob.multiplication(n1,n2);
	        break;
	case 4:ob.division(n1,n2);
	        break;
	default:System.out.println("Invalid choice");
	} }}

RESULT:

The above program is successfully executed and the output is obtained.

OUTPUT:



```
run:
Enter the 2 numbers:
5
5

1.Addition
2.Subtraction
3.Multiplication
4.Division
Enter the choice:
3
5.0 x 5.0 = 25.0
BUILD SUCCESSFUL (total time: 5 seconds)
```

PROGRAM NO-19

AIM:

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

ALGORITHM:

Step 1: Start the program.

Step 2: Create a User-defined exception named authException.

Step 3: Check for the validity of the username and the password against the required parameters. If any parameter is not met, then throw the authentication failed exception.

Step 6: To authenticate the credentials provided, check if the passwords and usernames given and entered are matching. If they match, display login successful, otherwise throw an exception.

Step 7: Stop the program.

CODE:

CO4Q3.java

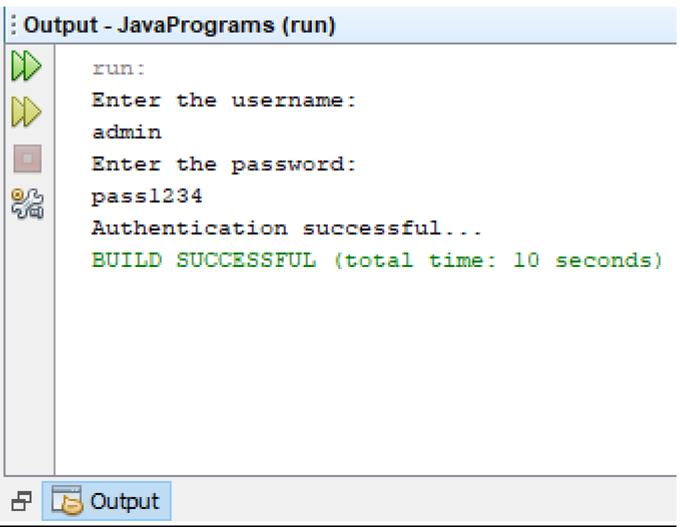
```
package javaprograms;
//Write a user defined exception class to authenticate the user name and
password
import java.util.Scanner;
class authException extends Exception{
    public authException(String s) {
        super(s);
    }


}
```

```
public class CO4Q3
{
    public static void main(String[] args) {
        String username = "admin";
        String passcode = "pass1234";
        String user_name,password;
        Scanner sc = new Scanner(System.in);
        try
        {
            System.out.println("Enter the username:");
            user_name = sc.nextLine();
//              sc.nextLine();
            System.out.println("Enter the password:");
            password = sc.nextLine();
            if(username.equals(user_name) && passcode.equals(password))
            {
                System.out.println("Authentication successful...");
            }
            else
                throw new authException("Invalid user credentials");

        }
        catch(authException e)
        {
            System.out.println("Exception caught "+e);
        }
    }

}
```

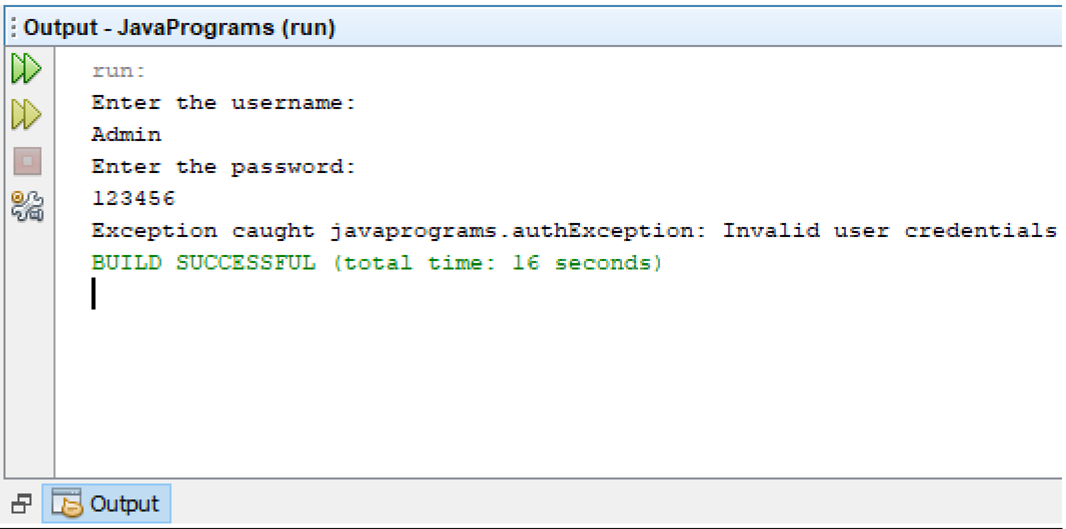
RESULT:

The above program is successfully executed and the output is obtained.

OUTPUT:



```
run:
Enter the username:
admin
Enter the password:
pass1234
Authentication successful...
BUILD SUCCESSFUL (total time: 10 seconds)
```



```
run:
Enter the username:
Admin
Enter the password:
123456
Exception caught javaprograms.authException: Invalid user credentials
BUILD SUCCESSFUL (total time: 16 seconds)
```

PROGRAM NO-20

AIM:

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

ALGORITHM:

Step 1: Start the program.

Step 2: Define a class named ‘*NegativeIntegerException*’ which throws a negative number exception.

Step 3: Get user inputs for integers at run time and throw an exception if the entered number is negative.

Step 4: Else, proceed to calculate the average and display the result.

Step 5: Stop the program.

CODE:

CO4Q4.java

```
package javaprograms;
//Find the average of N positive integers, raising a user defined exception for each negative
//input


import java.util.Scanner;


class NegativeIntegerException extends Exception{
    public NegativeIntegerException(String s){
        super(s);
    }
```

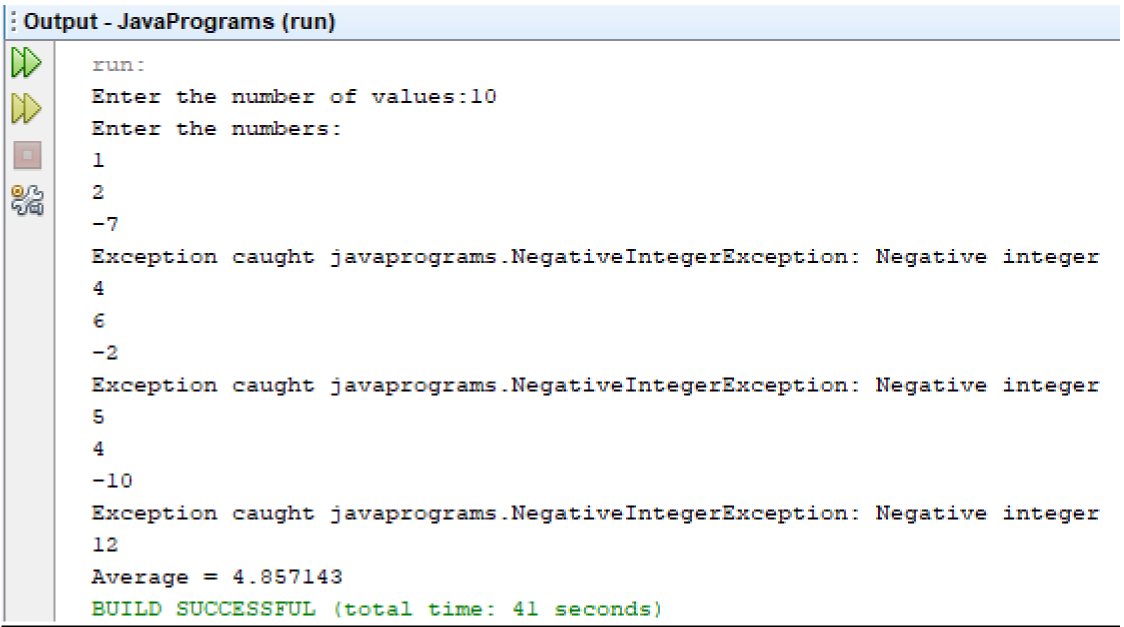
```
}
public class CO4Q4 {
    public static void sample(){
        try {
            int n,count=0;
            float num[];
            float total=0;
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter the number of values:");
            n = sc.nextInt();
            num = new float[n];
            System.out.println("Enter the numbers:");
            for(int i=0;i<n;i++){
                num[i] = sc.nextInt();
                try{
                if(num[i]<0){
                    throw new NegativeIntegerException("Negative integer");
                }
                else{
                    total += num[i];
                    count++;
                }
                }catch(NegativeIntegerException e)
                {
                    System.out.println("Exception caught "+e);
                }
            }
            System.out.println("Average = "+(total/count));
        } catch (Exception e) {
            System.out.println("Exception caught "+e);
        }}
```

	public static void main(String[] args) {
	    try {
	        sample();
	    } catch (Exception e) {
	    }}}

RESULT:

The above program is successfully executed and the output is obtained.

OUTPUT:



Output - JavaPrograms (run)

```
run:
Enter the number of values:10
Enter the numbers:
1
2
-7
Exception caught javaprograms.NegativeIntegerException: Negative integer
4
6
-2
Exception caught javaprograms.NegativeIntegerException: Negative integer
5
4
-10
Exception caught javaprograms.NegativeIntegerException: Negative integer
12
Average = 4.857143
BUILD SUCCESSFUL (total time: 41 seconds)
```

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 the program.

Step 2: Define classes named ‘*MultiplicationTable*’ and ‘*PrimeNumbers*’ that extends *Thread class* and contain methods to compute the multiplication table of 5 and to generate first N prime prime numbers respectively.

Step 3: Define a main method to create objects for the classes and invoke the associated methods.

Step 4: Stop the program.

CODE:

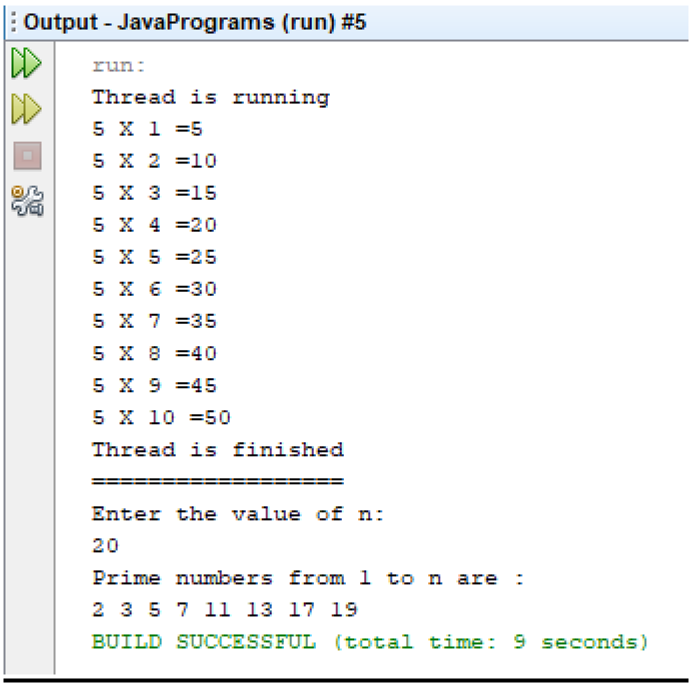
CO4Q5.java	
	```
	//Define 2 classes; one for generating multiplication table of 5 and other for displaying first
	//N prime numbers. Implement using threads. (Thread class)
	package javaprograms;

	import java.util.Scanner;
	import java.util.logging.Level;
	import java.util.logging.Logger;

	class MultiplicationTable extends Thread
	{
	    public void run()
	    {
	        System.out.println("Thread is running");
	        for(int i=1;i<=10;i++)
	        {
	            try
	```

```
            {
                System.out.println("5 X "+i+" ="+(5*i));
                Thread.sleep(200);
            } catch (InterruptedException ex) {

            }
        }
        System.out.println("Thread is finished");
        System.out.println("=================");
    }
}
class PrimeNumbers extends Thread
{
    public void run()
    {
      Scanner sc = new Scanner(System.in);
      int i =0;
      int num =0;
      String  primeNumbers = "";
      System.out.println("Enter the value of n:");
      int n = sc.nextInt();
      for (i = 1; i <= n; i++)
      {
         int counter=0;
         for(num =i; num>=1; num--)
         {
                if(i%num==0)
                {
                        counter = counter + 1;
                }
             }
             if (counter ==2)
             {
                primeNumbers = primeNumbers + i + " ";
             }
      }
      System.out.println("Prime numbers from 1 to n are :");
      System.out.println(primeNumbers);
    }
}
public class CO4Q5 {
    public static void main(String[] args) throws InterruptedException {
        MultiplicationTable MTOb = new MultiplicationTable();
        MTOb.start();
        Thread.sleep(5000);
        PrimeNumbers PNob = new PrimeNumbers();
        PNob.start();
    }
}
```

RESULT:

The above program is successfully executed and the output is obtained.

OUTPUT:


Output - JavaPrograms (run) #5

```
run:
Thread is running
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
Thread is finished
==================
Enter the value of n:
20
Prime numbers from 1 to n are :
2 3 5 7 11 13 17 19
BUILD SUCCESSFUL (total time: 9 seconds)
```

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 the program.

Step.2: Define classes named ‘*Fibonacci*’ and ‘*Even*’ that implements *Runnable interface* and contain methods to generate Fibonacci series and to display even numbers in a given range respectively.

Step.3: Define a main method to create objects for the classes and invoke the associated methods.

Step.4: Stop the program.

CODE:

CO4Q6.java	
	```
	//Define 2 classes; one for generating Fibonacci numbers and other for displaying even
	//numbers in a given range. Implement using threads. (Runnable Interface)
	package javaprograms;

	import java.util.Scanner;

	class Fibonacci implements Runnable
	{
	    int n,first,second,t;
	    String str;

	    public Fibonacci(int num)
	    {
	```

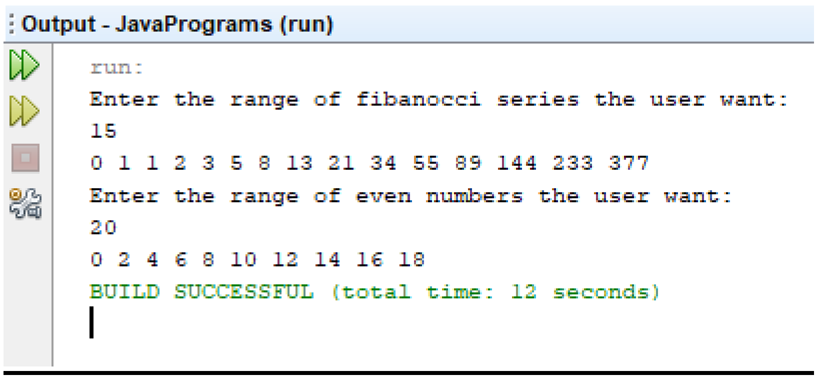
```
        n = num;
        first = 0;
        second = 1;
    }

    @Override
    public void run()
    {
        str = first+" "+second;
        for(int i=0;i<=n-3;i++)
        {
            t = first + second;
            first = second;
            second = t;
            str += " "+t;
        }
        System.out.println(str);
    }

}
class Even implements Runnable
{
    int n;
    String str;
    public Even(int n)
    {
        this.n = n;
        str = "";
    }
    @Override
    public void run()
    {
        for(int i=0;i<n;i=i+2)
            if(i%2==0)
            {
                str+=i+" ";
            }
        System.out.println(str);
    }

}
public class CO4Q6 {
    public static void main(String[] args) throws InterruptedException {
        int n1,n2;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the range of fibanocci series the user
want:");
        n1 = sc.nextInt();
        Fibonacci fibob = new Fibonacci(n1);
```

	```
	        Thread th = new Thread(fibob);
	        th.start();
	        Thread.sleep(400);
	        System.out.println("Enter the range of even numbers the user want:");
	        n2 = sc.nextInt();
	        Even evob = new Even(n2);
	        Thread th2 = new Thread(evob);
	        th2.start();
	    }
	}
	```

RESULT:

The above program is successfully executed and the output is obtained.

OUTPUT:



Output - JavaPrograms (run)

```
run:
Enter the range of fibanocci series the user want:
15
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377
Enter the range of even numbers the user want:
20
0 2 4 6 8 10 12 14 16 18
BUILD SUCCESSFUL (total time: 12 seconds)
```

PROGRAM NO-23

AIM:

Producer/Consumer using ITC

ALGORITHM:

Step 1: Start the program.

Step 2: Define a class that contains two threads that will simulate producer and consumer.

Step 3: Define a class ‘Producer’ and ‘Consumer’ which will contain a LinkedList of integers.

Step 4: Define a method produce() that produces items till its capacity is reached and notify the consumer thread to start consuming.

Step 5: Define a method consume() that consumes items till the list is empty and notify the producer thread to start producing.

Step 6: Stop the program.

CODE:

CO4Q7.java

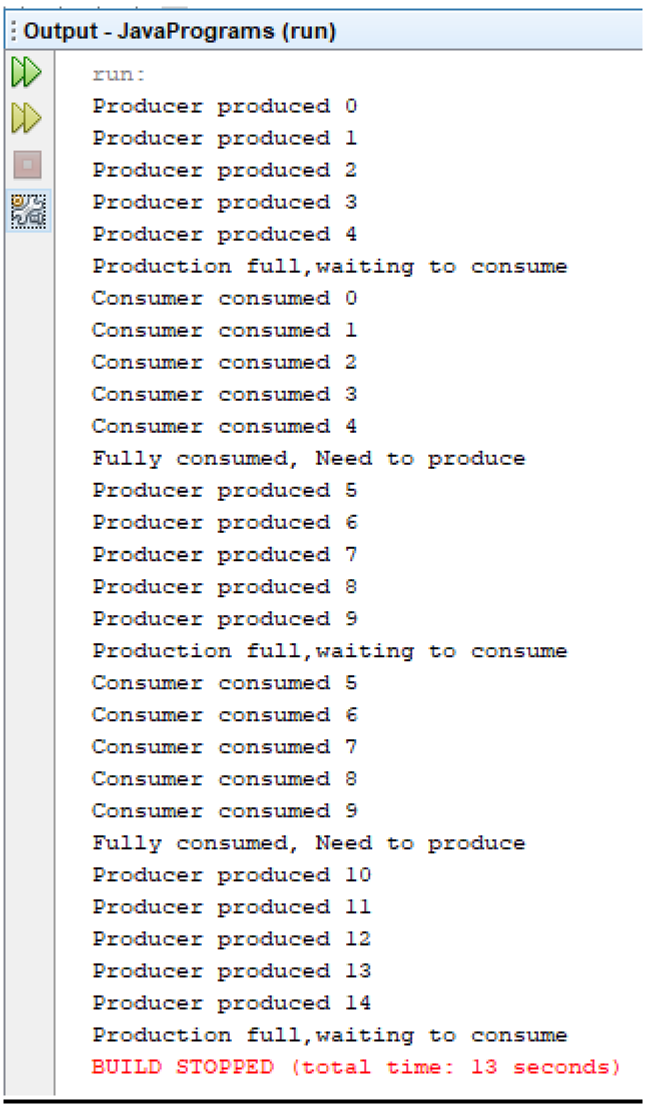
```
package javaprograms;
//Producer/Consumer using ITC

import java.util.ArrayList;
import java.util.List;

class Producer implements Runnable
{
    List<Integer> flist;
    int max_size = 5;
    int i=0;
    Producer(List<Integer> flist)
    {
        this.flist = flist;
    }
    @Override
    public void run()
    {
        while(true)
        {
            try
            {
                produce(i++);
            } catch (Exception e)
            {
                System.out.println("Intteruption "+e);
            }
        }
    }
    public void produce(int i) throws InterruptedException
    {
        synchronized (flist)
        {
            while(flist.size()==max_size)
            {
                System.out.println("Production full,waiting to consume");
                flist.wait();
            }
        }
        synchronized(flist)
        {
            System.out.println("Producer produced "+i);
            flist.add(i);
            flist.notify();
        }
    }

}
class Consumer implements Runnable
{
    List<Integer> flist;
    Consumer(List<Integer> flist)
    {
        this.flist = flist;
    }

    @Override
    public void run()
    {
        while(true)
        {
            try
            {
                consume();
            } catch (Exception e)
            {
```

```
            System.out.println("Exception "+e);
          }
        }
    }
    public void consume() throws InterruptedException
    {
        synchronized (flist)
        {
          while(flist.isEmpty())
          {
            System.out.println("Fully consumed, Need to produce");
            flist.notify();
            Thread.sleep(500);
            flist.wait();
          }
        }
        synchronized(flist)
        {
          Thread.sleep(1000);
          System.out.println("Consumer consumed "+flist.remove(0));
        }
    }

}
public class CO4Q7 {
    public static void main(String[] args)
    {
        List<Integer> flist = new ArrayList<Integer>();
        Thread th1 = new Thread(new Producer(flist));
        Thread th2 = new Thread(new Consumer(flist));
        th1.start();
        th2.start();
    }
}
```

RESULT:

The above program is successfully executed and the output is obtained.

OUTPUT:


Output - JavaPrograms (run)

```
run:
Producer produced 0
Producer produced 1
Producer produced 2
Producer produced 3
Producer produced 4
Production full,waiting to consume
Consumer consumed 0
Consumer consumed 1
Consumer consumed 2
Consumer consumed 3
Consumer consumed 4
Fully consumed, Need to produce
Producer produced 5
Producer produced 6
Producer produced 7
Producer produced 8
Producer produced 9
Production full,waiting to consume
Consumer consumed 5
Consumer consumed 6
Consumer consumed 7
Consumer consumed 8
Consumer consumed 9
Fully consumed, Need to produce
Producer produced 10
Producer produced 11
Producer produced 12
Producer produced 13
Producer produced 14
Production full,waiting to consume
BUILD STOPPED (total time: 13 seconds)
```

PROGRAM NO-24

AIM:

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

ALGORITHM:

Step 1: Start the program.

Step 2: Define a class '*StackElement*' that contains methods to push, pop and display the stack elements.

Step 4: Display the results.

Step 5: Stop the program.

CODE:

CO4Q8.java	code

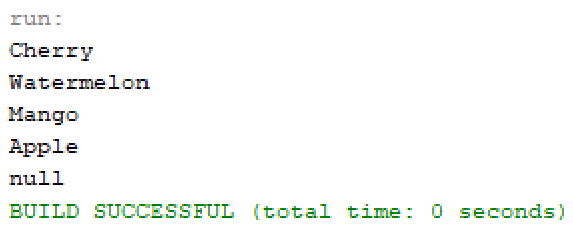
```
class StackElement<T>{
    T value;
    StackElement<T> next;
    public StackElement(T value,StackElement<T> next) {
        this.value = value;
        this.next = next;
    }
    public StackElement<T> getNext(){
        return next;
    }
    public T getValue(){
        return value;
    }}
public class CO4Q8 <T>{
    int size;
    StackElement<T> top;
    public CO4Q8(){
        size = 0;
        top = null;
    }
    public void push(T newValue){
```

	```
	        StackElement<T> newElement = new
	StackElement<T>(newValue,top);
	        top = newElement;
	        size++;
	    }
	    public T pop()
	    {
	        StackElement<T> oldTop = top;
	        if(size==0)
	        {
	            return null;
	        }
	        top = top.getNext();
	        size--;
	        return oldTop.getValue();
	    }

	    public static void main(String[] args)
	    {
	        CO4Q8 <String> strStack = new CO4Q8<String>();
	        strStack.push("Apple");
	        strStack.push("Mango");
	        strStack.push("Watermelon");
	        strStack.push("Cherry");
	        System.out.println(strStack.pop());
	        System.out.println(strStack.pop());
	        System.out.println(strStack.pop());
	        System.out.println(strStack.pop());
	        System.out.println(strStack.pop());
	    }
	}
	```

RESULT:

The above program is successfully executed and the output is obtained.

OUTPUT:



```
run:
Cherry
Watermelon
Mango
Apple
null
BUILD SUCCESSFUL (total time: 0 seconds)
```

PROGRAM NO-25

AIM:

Using generic method, perform Bubble sort.

ALGORITHM:

Step 1: Start the program.

Step 2: Define a class using generics

Step 3: Define a function '*bubblesort*'.

Step 6: Read the numbers and perform bubblesort.

Step 7: Stop the program.

CODE:

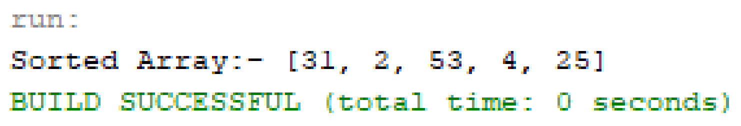
CO4Q9.java

```
import java.util.*;
public class CO4Q9<T extends Comparable<? super T>>
{
    T[] array;
    CO4Q9(T[] array)
    {
        this.array = array;
    }
    public T[] bubbleSort()
    {
        for(int i = array.length; i>1; i--)
        {
            for(int j=0; j<i-1; j++)
            {
                if(array[j].compareTo(array[j+1])>0)
                {
                    swap(j,array);
                }
            }
        }
        return array;
    }
    public void swap(int index, T[] arr)
    {
        T temp = arr[index];
        arr[index] = arr[index+1];
        arr[index] = temp;
    }
    public static void main(String[] args)
    {
        Integer[] intArr = {31,2,53,4,25};
        CO4Q9<Integer> BubbleSort = new CO4Q9<Integer>(intArr);
        Integer[] SortedArray = BubbleSort.bubbleSort();
        System.out.println("Sorted Array:- "+Arrays.toString(SortedArray));
    }
}
```

RESULT:

The above program is successfully executed and the output is obtained.

OUTPUT:



```
run:
Sorted Array:- [31, 2, 53, 4, 25]
BUILD SUCCESSFUL (total time: 0 seconds)
```

PROGRAM NO-26

AIM:

To maintain a list of Strings using ArrayList from collection framework, perform built-in operations.

ALGORITHM:

Step 1: Start the program.

Step 2: Define a class with a main() to implement the concept.

Step 3: Declare an ArrayList of strings named ‘*fruits*’ and start adding elements into it.

Step 4: Execute different ArrayList methods and display the results.

Step 5: Stop the program.

CODE:

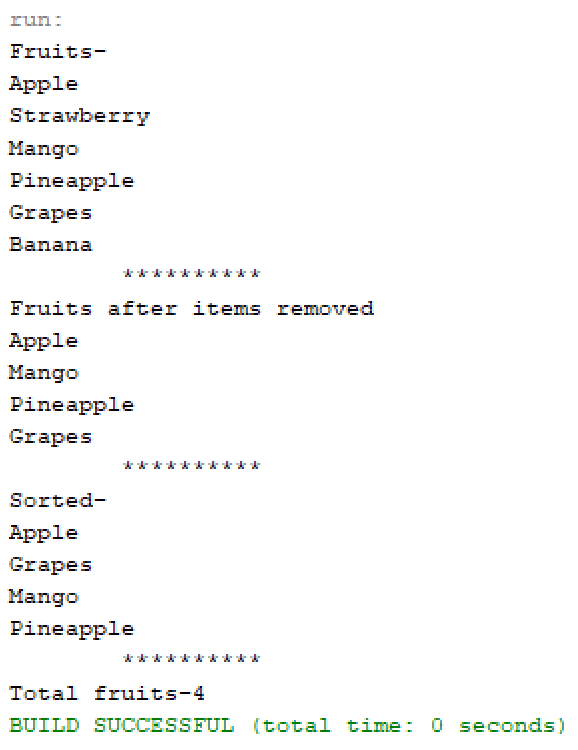
CO4Q10.java

```
import java.util.*;

public class CO4Q10 {
    public static void main(String[] args) {
        ArrayList<String> fruits = new ArrayList<String>();
        fruits.add("Apple");
        fruits.add("Strawberry");
            fruits.add("Mango");
            fruits.add("Pineapple");
            fruits.add("Banana");
            fruits.add(4, "Grapes");
        System.out.println("Fruits-");
        for(String str: fruits)
            System.out.println(str+" ");
        fruits.remove("Banana");
        fruits.remove(1);
        System.out.println("\t**********");
        System.out.println("Fruits after items removed");
        for(String str: fruits)
            System.out.println(str+" ");
        Collections.sort(fruits);
        System.out.println("\t**********");
        System.out.println("Sorted-");
        for(String str: fruits)
            System.out.println(str+" ");
        System.out.println("\t**********");
        System.out.println("Total fruits-"+fruits.size());
    }
}
```

RESULT:

The above program is successfully executed and the output is obtained.

OUTPUT:



```
run:
Fruits-
Apple
Strawberry
Mango
Pineapple
Grapes
Banana
        **********
Fruits after items removed
Apple
Mango
Pineapple
Grapes
        **********
Sorted-
Apple
Grapes
Mango
Pineapple
        **********
Total fruits-4
BUILD SUCCESSFUL (total time: 0 seconds)
```

PROGRAM NO-27

AIM:

To write a program to remove all the elements from a linked list.

ALGORITHM:

Step 1: Start the program.

Step 2: Define a class with a main() to implement the concept.

Step 3: Create a LinkedList.

Step 4: Using add() to add to LinkedList

Step 5: Using clear() to remove all from the LinkedList

Step 6: Stop the program.

CODE:

CO4Q11.java

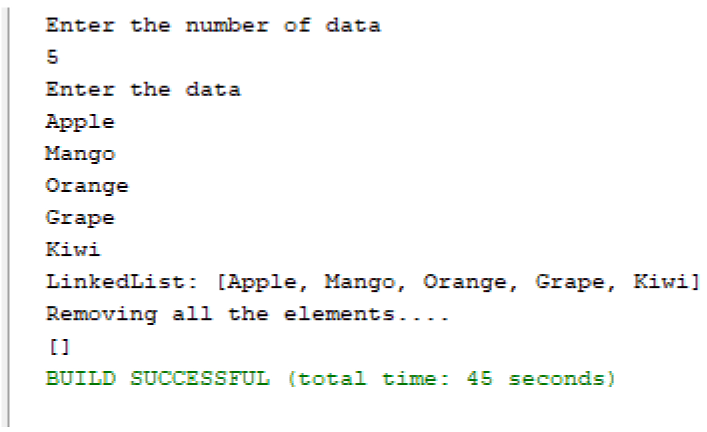
```
import java.util.LinkedList;
import java.util.Scanner;
public class CO4Q11
{

    public static void main(String[] args) {
        int n;
        String data;
        LinkedList<String> ll = new LinkedList<String>();
        System.out.println("Enter the number of data");
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        System.out.println("Enter the data");
        sc.nextLine();
        for(int i=0;i<n;i++)
        {
            data = sc.nextLine();
            ll.add(data);
```

	`}`
	`System.out.println("LinkedList: "+ll);`
	`System.out.println("Removing all the elements....");`
	`ll.clear();`
	`System.out.println(ll);`
	`}`
	`}`

RESULT:

The above program is successfully executed and the output is obtained.

OUTPUT:



```
Enter the number of data
5
Enter the data
Apple
Mango
Orange
Grape
Kiwi
LinkedList: [Apple, Mango, Orange, Grape, Kiwi]
Removing all the elements....
[]
BUILD SUCCESSFUL (total time: 45 seconds)
```

PROGRAM NO-28

AIM:

To write a program to remove an object from the Stack when the position is passed as parameter.

ALGORITHM:

Step 1: Start the program.

Step 2: Define a class with a main () to implement the concept.

Step 3: Declare a Stack of Strings and start adding elements into it using add() method.

Step 5: Using remove(index) method, remove the element and display the updated stack.

Step 6: Stop the program.

CODE:

CO4Q12.java

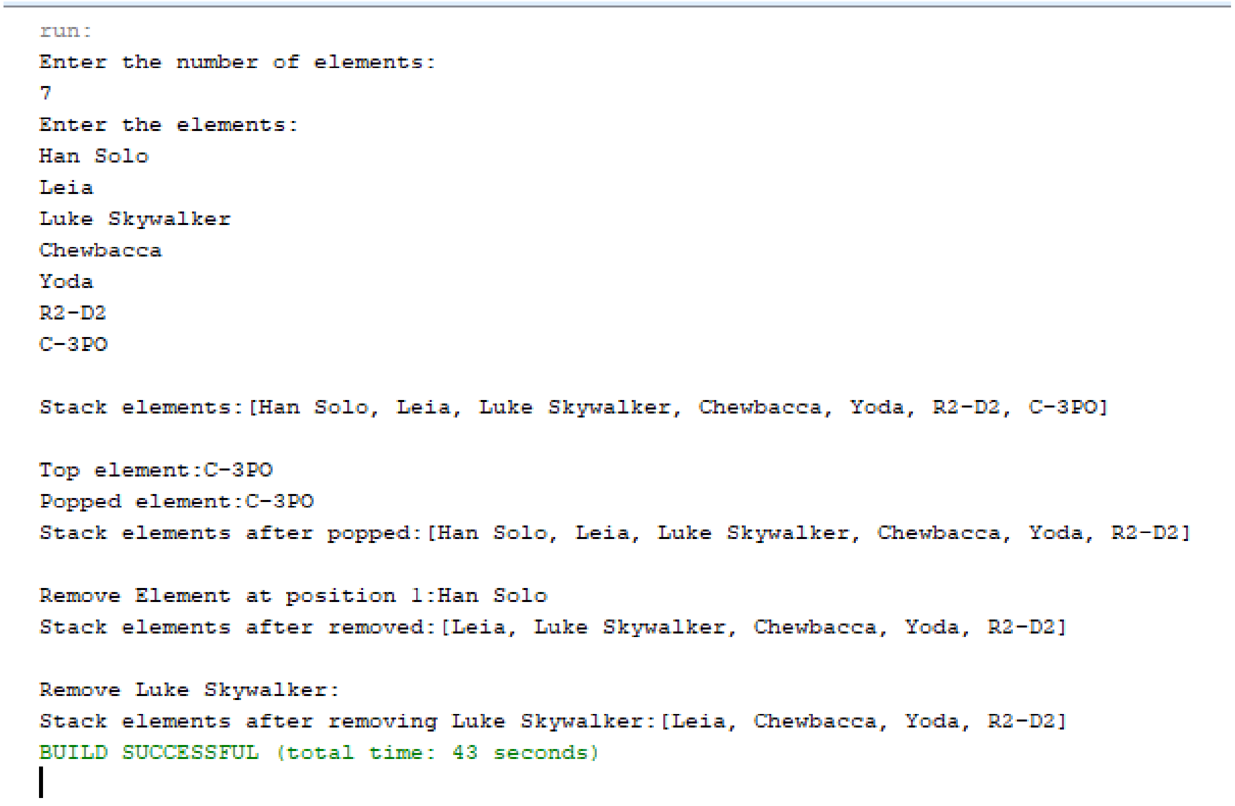
```
import java.util.Scanner;
import java.util.Stack;

public class CO4Q12 {
    public static void main(String[] args) {
        int n;
        String str;
        Stack<String> s = new Stack<String>();
        System.out.println("Enter the number of elements:");
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        sc.nextLine();
        System.out.println("Enter the elements:");
        for(int i=0;i<n;i++)
        {
            str = sc.nextLine();
            s.add(str);
        }
        System.out.println("\nStack elements:"+s);
```

	```
	        System.out.println("\nTop element:"+s.peek());
	        System.out.println("Popped element:"+s.pop());
	        System.out.println("Stack elements after popped:"+s);
	        System.out.println("\nRemove Element at position 1:"+s.remove(0));
	        System.out.println("Stack elements after removed:"+s);
	        System.out.println("\nRemove Luke Skywalker:");
	        s.remove("Luke Skywalker");
	        System.out.println("Stack elements after removing Luke
	Skywalker:"+s);
	    }
	}
	```

RESULT:

The above program is successfully executed and the output is obtained.

OUTPUT:



```
run:
Enter the number of elements:
7
Enter the elements:
Han Solo
Leia
Luke Skywalker
Chewbacca
Yoda
R2-D2
C-3PO

Stack elements:[Han Solo, Leia, Luke Skywalker, Chewbacca, Yoda, R2-D2, C-3PO]

Top element:C-3PO
Popped element:C-3PO
Stack elements after popped:[Han Solo, Leia, Luke Skywalker, Chewbacca, Yoda, R2-D2]

Remove Element at position 1:Han Solo
Stack elements after removed:[Leia, Luke Skywalker, Chewbacca, Yoda, R2-D2]

Remove Luke Skywalker:
Stack elements after removing Luke Skywalker:[Leia, Chewbacca, Yoda, R2-D2]
BUILD SUCCESSFUL (total time: 43 seconds)
```

PROGRAM NO-29

AIM:

To write a program to demonstrate the creation of queue object using the PriorityQueue class.

ALGORITHM:

Step 1: Start the program.

Step 2: Define a class with a main () to implement the concept.

Step 3: Declare a *PriorityQueue* of Strings and start adding elements into it using *add()* method.

Step 4: Iterate and display the queue elements.

Step 5: Stop the program.

CODE:

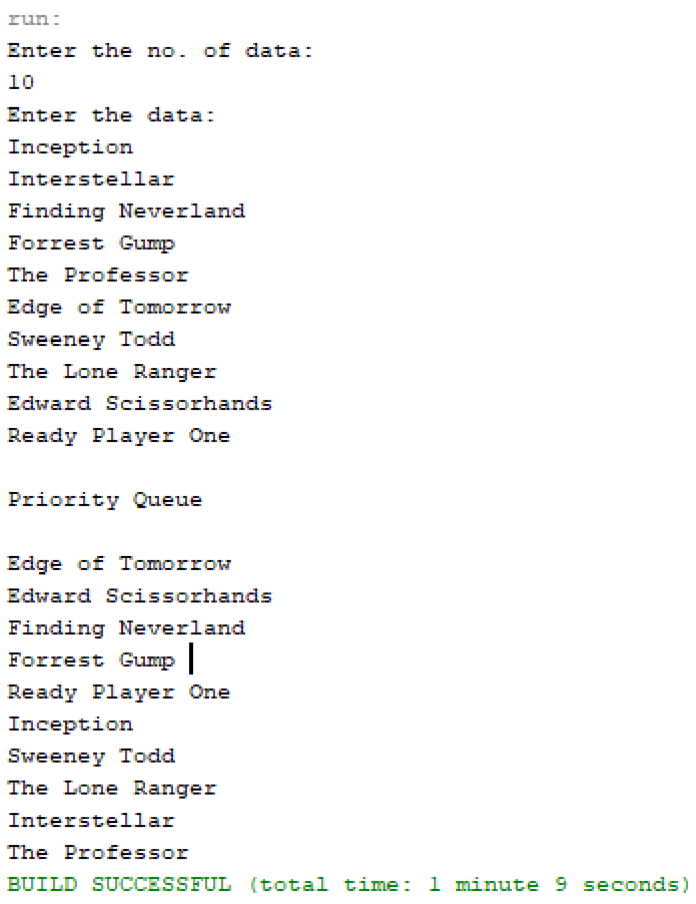
CO4Q13.java

```
import java.util.Iterator;
import java.util.PriorityQueue;
import java.util.Scanner;

public class CO4Q13 {
    public static void main(String[] args) {
        int n;
        String str;
        PriorityQueue<String> pq = new PriorityQueue<String>();
        System.out.println("Enter the no. of data:");
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        sc.nextLine();
        System.out.println("Enter the data:");
        for(int i=0;i<n;i++)
        {
            str = sc.nextLine();
            pq.add(str);
        }
        Iterator itr = pq.iterator();
        System.out.println("\nPriority Queue\n");
        while(itr.hasNext())
           System.out.println(itr.next()+" ");
    }
}
```

RESULT:

The above program is successfully executed and the output is obtained.

OUTPUT:



```
run:
Enter the no. of data:
10
Enter the data:
Inception
Interstellar
Finding Neverland
Forrest Gump
The Professor
Edge of Tomorrow
Sweeney Todd
The Lone Ranger
Edward Scissorhands
Ready Player One

Priority Queue

Edge of Tomorrow
Edward Scissorhands
Finding Neverland
Forrest Gump
Ready Player One
Inception
Sweeney Todd
The Lone Ranger
Interstellar
The Professor
BUILD SUCCESSFUL (total time: 1 minute 9 seconds)
```

PROGRAM NO-30

AIM:

To write a program to demonstrate the addition and deletion of elements in deque.

ALGORITHM:

Step 1: Start the program.

Step 2: Define a class with a main () to implement the concept.

Step 3: Declare a *Dequeue* of Strings and perform the following methods:

- addFirst() - inserts the element passed in the parameter to the front of the *Deque* if there is space.
- addLast() - inserts the element passed in the parameter to the end of the *Deque* if there is space.
- removeFirst() - removes the first element of Deque.
- removeLast()- removes the last element of Deque.

Step 4: Display the results.

Step 5: Stop the program.

CODE:

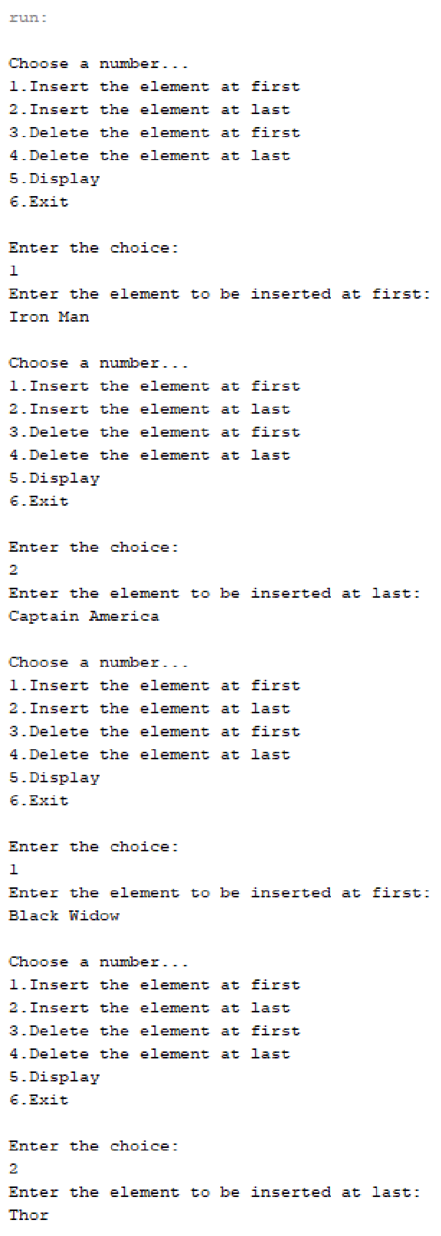
CO4Q14.java	
	```
	import java.util.Deque;
	import java.util.LinkedList;
	import java.util.Scanner;

	public class CO4Q14 {
	    public static void main(String[] args) {
	        int ch;
	        String data;
	        Deque<String> dq = new LinkedList<String>();
	        Scanner sc = new Scanner(System.in);
	        do
	```

	```
	        {
	           System.out.println("\nChoose a number...");
	           System.out.println("1.Insert the element at first");
	           System.out.println("2.Insert the element at last");
	           System.out.println("3.Delete the element at first");
	           System.out.println("4.Delete the element at last");
	           System.out.println("5.Display");
	           System.out.println("6.Exit");
	           System.out.println("\nEnter the choice:");
	           ch = sc.nextInt();
	           sc.nextLine();
	           switch(ch)
	           {
	              case 1: System.out.println("Enter the element to be inserted at
	first:");
	                    data = sc.nextLine();
	                    dq.addFirst(data);
	                    break;
	              case 2: System.out.println("Enter the element to be inserted at
	last:");
	                    data = sc.nextLine();
	                    dq.addLast(data);
	                    break;
	              case 3: System.out.println("Element deleted from the first
	position");
	                    dq.removeFirst();
	                    break;
	              case 4: System.out.println("Element deleted from the last
	position");
	                    dq.removeLast();
	                    break;
	              case 5: System.out.println("Elements:");
	                    System.out.println(dq);
	                    break;
	              case 6: System.exit(0);
	                    break;
	              default:System.out.println("Invalid choice...");
	           }
	        }while(true);
	    }
	}
	```

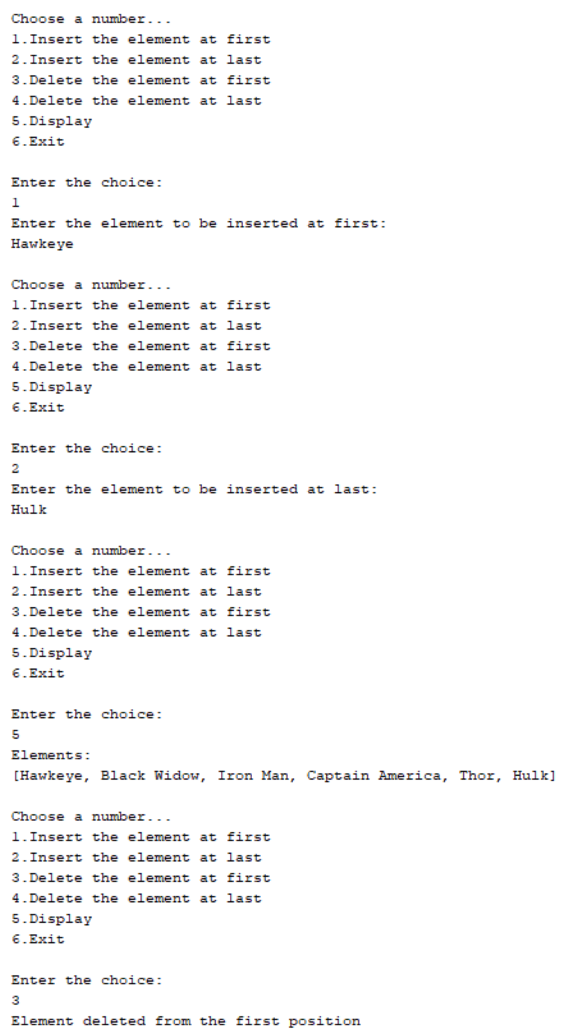
RESULT:

The above program is successfully executed and the output is obtained.

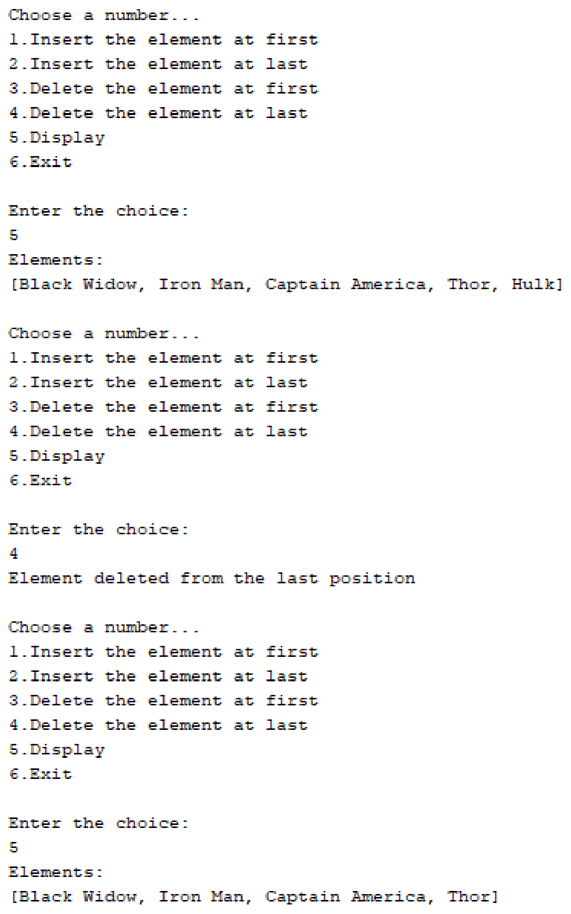
OUTPUT:



```
run:

Choose a number...
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
Enter the element to be inserted at first:
Iron Man

Choose a number...
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:
2
Enter the element to be inserted at last:
Captain America

Choose a number...
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
Enter the element to be inserted at first:
Black Widow

Choose a number...
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:
2
Enter the element to be inserted at last:
Thor
```



```
Choose a number...
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
Enter the element to be inserted at first:
Hawkeye

Choose a number...
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:
2
Enter the element to be inserted at last:
Hulk

Choose a number...
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:
5
Elements:
[Hawkeye, Black Widow, Iron Man, Captain America, Thor, Hulk]

Choose a number...
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:
3
Element deleted from the first position
```



```
Choose a number...
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:
5
Elements:
[Black Widow, Iron Man, Captain America, Thor, Hulk]

Choose a number...
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:
4
Element deleted from the last position

Choose a number...
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:
5
Elements:
[Black Widow, Iron Man, Captain America, Thor]
```

PROGRAM NO-31

AIM:

To write a program to demonstrate the creation of Set object using the LinkedHashset class.

ALGORITHM:

Step 1: Start the program.

Step 2: Define a class with a main () to implement the concept.

Step 3: Declare an *LinkedHashSet* of strings.

Step 4: Start adding elements into it using add() method.

Step 5: Execute some LinkedHashSet methods and display the results.

Step 6: Stop the program.

CODE:

CO4Q15.java

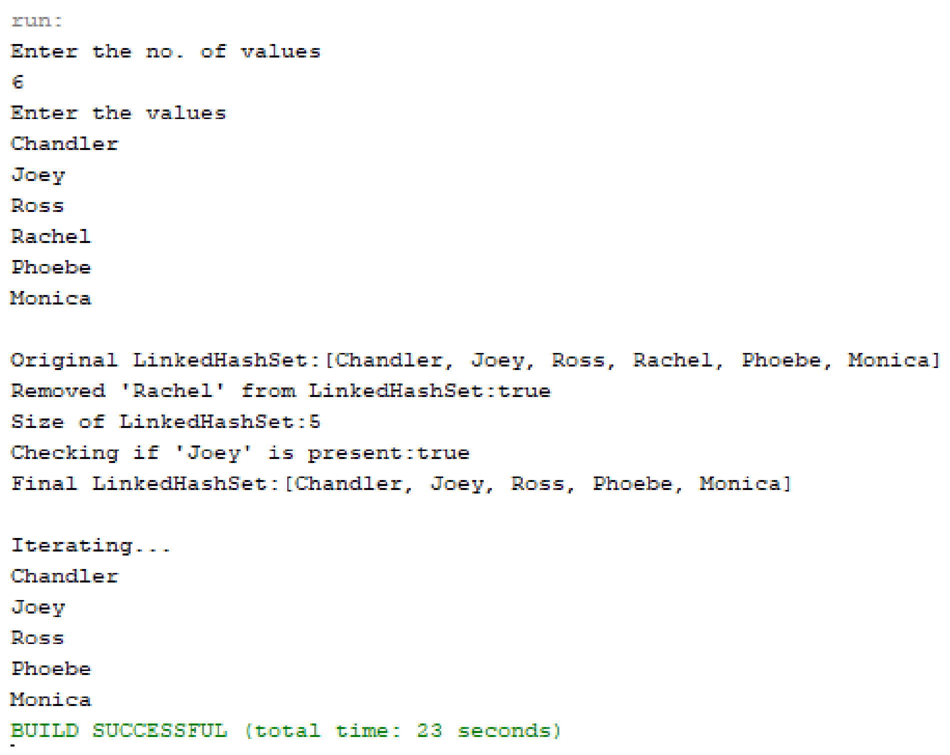
```
import java.util.Iterator;
import java.util.LinkedHashSet;
import java.util.Scanner;

public class CO4Q15 {
    public static void main(String[] args) {
        int n;
        String str;
        LinkedHashSet<String> lsh = new LinkedHashSet<String>();
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the no. of values");
        n = sc.nextInt();
        sc.nextLine();
        System.out.println("Enter the values");
        for(int i=0;i<n;i++)
        {
            str = sc.nextLine();
            lsh.add(str);
        }
```

	```
	        System.out.println("\nOriginal LinkedHashSet:"+lsh);
	        System.out.println("Removed 'Rachel' from
	LinkedHashSet:"+lsh.remove("Rachel"));
	        System.out.println("Size of LinkedHashSet:"+lsh.size());
	        System.out.println("Checking if 'Joey' is
	present:"+lsh.contains("Joey"));
	        System.out.println("Final LinkedHashSet:"+lsh);
	        System.out.println("\nIterating...");
	        Iterator itr = lsh.iterator();
	        while(itr.hasNext())
	            System.out.println(itr.next());

	    }

	}
	```

RESULT:

The above program is successfully executed and the output is obtained.

OUTPUT:



```
run:
Enter the no. of values
6
Enter the values
Chandler
Joey
Ross
Rachel
Phoebe
Monica

Original LinkedHashSet:[Chandler, Joey, Ross, Rachel, Phoebe, Monica]
Removed 'Rachel' from LinkedHashSet:true
Size of LinkedHashSet:5
Checking if 'Joey' is present:true
Final LinkedHashSet:[Chandler, Joey, Ross, Phoebe, Monica]

Iterating...
Chandler
Joey
Ross
Phoebe
Monica
BUILD SUCCESSFUL (total time: 23 seconds)
```

PROGRAM NO-32

AIM:

To write a Java program to compare two hash sets.

ALGORITHM:

Step 1: Start the program.

Step 2: Define a class with a main () to implement the concept.

Step 3: Declare two HashSets (s1, s2) of strings and start adding elements into them using add() method.

Step 4: Display the elements of both the hashsets.

Step 5: Perform set operations.

- addAll() – gets all the elements
- retainAll() – gets the elements which are common
- removeAll() – gets the difference

Step 6: Display the results.

Step 7: Stop the program.

CODE:

CO4Q16.java

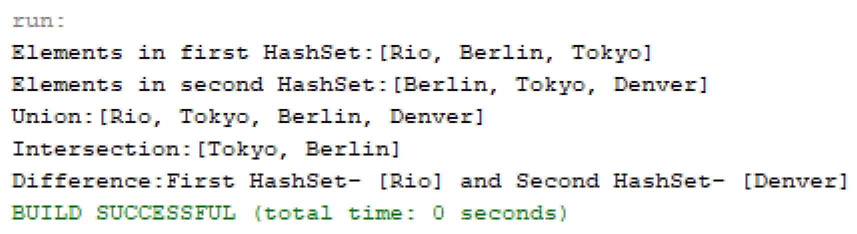
```
import java.util.*;

public class CO4Q16 {
    public static void union(Set<String> s1,Set<String> s2)
    {
        Set<String> su = new HashSet<>(s1);
        su.addAll(s2);
        System.out.println("Union:"+su);
    }
    public static void intersection(Set<String> s1,Set<String> s2)
    {
        Set<String> su = new HashSet<>(s1);
        su.retainAll(s2);
```

	```
	        System.out.println("Intersection:"+su);
	    }
	    public static void difference(Set<String> s1,Set<String> s2)
	    {
	        Set<String> su1 = new HashSet<>(s1);
	        su1.removeAll(s2);
	        Set<String> su2 = new HashSet<>(s2);
	        su2.removeAll(s1);
	        System.out.println("Difference:First HashSet- "+su1+" and Second HashSet- "+su2);
	    }
	    public static void main(String[] args) {
	        Set<String> s1 = new HashSet<>();
	        s1.add("Tokyo");
	        s1.add("Berlin");
	        s1.add("Rio");
	        Set<String> s2 = new HashSet<>();
	        s2.add("Tokyo");
	        s2.add("Berlin");
	        s2.add("Denver");
	        System.out.println("Elements in first HashSet:"+s1);
	        System.out.println("Elements in second HashSet:"+s2);
	        union(s1,s2);
	        intersection(s1,s2);
	        difference(s1,s2);
	    }
	}
	```

RESULT:

The above program is successfully executed and the output is obtained.

OUTPUT:



```
run:
Elements in first HashSet:[Rio, Berlin, Tokyo]
Elements in second HashSet:[Berlin, Tokyo, Denver]
Union:[Rio, Tokyo, Berlin, Denver]
Intersection:[Tokyo, Berlin]
Difference:First HashSet- [Rio] and Second HashSet- [Denver]
BUILD SUCCESSFUL (total time: 0 seconds)
```

PROGRAM NO-33

AIM:

To write a program to demonstrate the working of Map interface by adding, changing and removing elements.

ALGORITHM:

Step.1: Start the program.

Step.2: Define a class with a main () to implement the concept.

Step.3: Declare a *HashMap* and insert elements into it using *put()* method.

Step.4: To update any value in hashmap using *hashmap.put()* or *hashmap.replace()*.

Step.5: Using *remove(key)* method, remove elements from hashmap.

Step.6: Display the results.

Step.7: Stop the program.

CODE:

CO4Q17.java

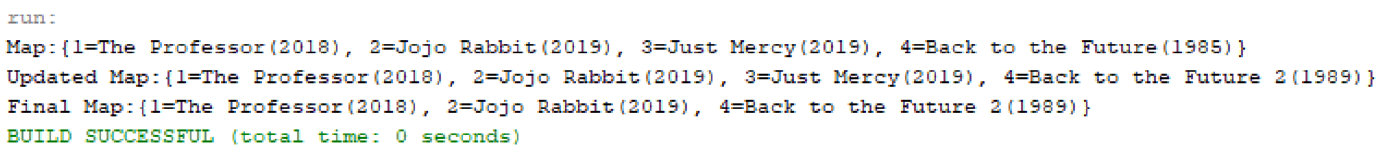
```
import java.util.HashMap;
import java.util.Map;

public class CO4Q17 {
    public static void main(String[] args) {
        Map<Integer, String> hm = new HashMap<>();
        hm.put(1,"The Professor(2018)");
        hm.put(2,"Jojo Rabbit(2019)");
        hm.put(3,"Just Mercy(2019)");
        hm.put(4,"Back to the Future(1985)");
        System.out.println("Map:"+hm);
        hm.put(4,"Back to the Future 2(1989)");
        System.out.println("Updated Map:"+hm);
        hm.remove(3);
        System.out.println("Final Map:"+hm);
```

	} }

RESULT:

The above program is successfully executed and the output is obtained.

OUTPUT:



```
run:
Map:{1=The Professor(2018), 2=Jojo Rabbit(2019), 3=Just Mercy(2019), 4=Back to the Future(1985)}
Updated Map:{1=The Professor(2018), 2=Jojo Rabbit(2019), 3=Just Mercy(2019), 4=Back to the Future 2(1989)}
Final Map:{1=The Professor(2018), 2=Jojo Rabbit(2019), 4=Back to the Future 2(1989)}
BUILD SUCCESSFUL (total time: 0 seconds)
```

PROGRAM NO-34

AIM:

To write a program to Convert HashMap to TreeMap.

ALGORITHM:

Step 1: Start the program.

Step 2: Define a class with a main () to implement the concept.

Step 3: Declare a *HashMap* and insert elements into it using *put()* method.

Step 4: Convert the above HashMap to TreeMap:

Map<i,s> treeMap = new TreeMap<>();

treeMap.putAll(hashMap);

Step 5: Display the resultant treeMap.

Step 6: Stop the program.

CODE:

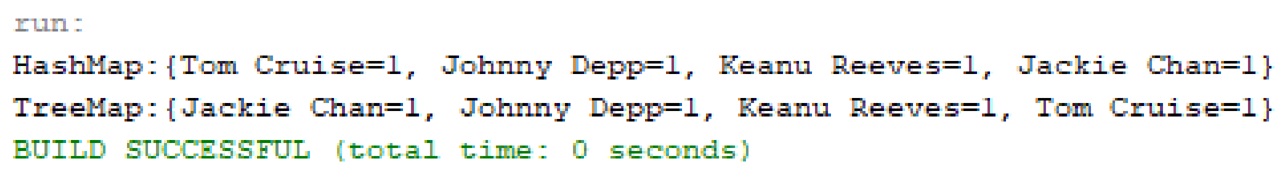
CO4Q18.java	
	```
	import java.util.*;

	public class CO4Q18 {
	    public static <i,s> Map<i,s> convert(Map<i,s> hashmap)
	    {
	        Map<i,s> treemap = new TreeMap<>();
	        treemap.putAll(hashmap);
	        return treemap;
	    }
	    public static void main(String[] args) {
	        Map<String,Integer> hashmap = new HashMap<>();
	        hashmap.put("Johnny Depp",1);
	        hashmap.put("Tom Cruise",1);
	        hashmap.put("Jackie Chan",1);
	```

```
        hashmap.put("Keanu Reeves",1);
        System.out.println("HashMap:"+hashmap);
        Map<String,Integer>treemap = convert(hashmap);
        System.out.println("TreeMap:"+treemap);
    }
}
```

RESULT:

The above program is successfully executed and the output is obtained.

OUTPUT:



```
run:
HashMap:{Tom Cruise=1, Johnny Depp=1, Keanu Reeves=1, Jackie Chan=1}
TreeMap:{Jackie Chan=1, Johnny Depp=1, Keanu Reeves=1, Tom Cruise=1}
BUILD SUCCESSFUL (total time: 0 seconds)
```

PROGRAM NO: 35

AIM:

To write a program to draw Circle, Rectangle, Line in Applet.

ALGORITHM:

Step.1: Start the program.

Step.2: Define a class ‘CO5Q1’ that extends Applet class.

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

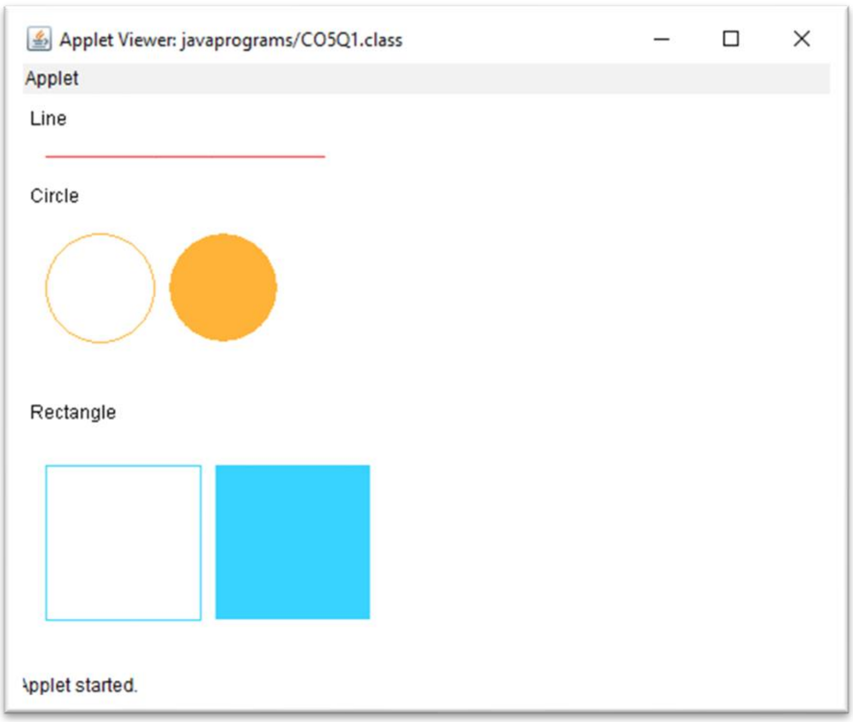
Step.4: Stop the program.

PROGRAM CODE:

CO5Q1.java

```
//Program to draw Circle, Rectangle, Line in Applet.
package javaprograms;

import java.applet.*;
import java.awt.*;


public class CO5Q1 extends Applet
{
   public void paint(Graphics g)
   {
      g.drawString("Line",10,20);
      g.setColor(Color.decode("#FF3333"));
      g.drawLine(20, 40, 200, 40);
      g.setColor(Color.black);
      g.drawString("Circle", 10, 70);
      g.setColor(Color.decode("#FFB033"));
      g.drawOval(20, 90, 70, 70);
      g.fillOval(100, 90, 70, 70);
      g.setColor(Color.black);
      g.drawString("Rectangle", 10, 210);
      g.setColor(Color.decode("#33D0FF"));
      g.drawRect(20, 240, 100, 100);
      g.fillRect(130, 240, 100, 100);
   }
}
```

OUTPUT:



RESULT:

The program is successfully executed and the output is verified.

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 ‘CO5Q2’ 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.

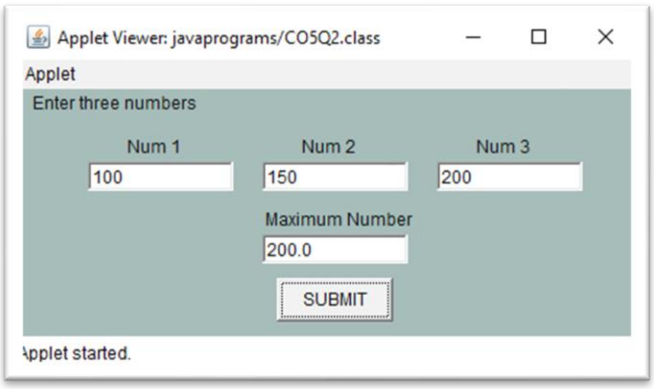
Step.8: Stop the program.

PROGRAM CODE:

CO5Q2.java

```
package javaprograms;
//Program to find maximum of three numbers using AWT.

import java.applet.Applet;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class CO5Q2 extends Applet implements ActionListener
{
        TextField t1,t2,t3,result;
        Button result_button;
        Label l1,l2,l3,result_label,title;
    public void init()
    {
        setLayout(null);
        setBackground(Color.decode("#A4BAB7"));
        setForeground(Color.decode("#080F0F"));
        t1 = new TextField();
        t2 = new TextField();
        t3 = new TextField();
        result = new TextField();
        l1 = new Label("Num 1");
        l2 = new Label("Num 2");
        l3 = new Label("Num 3");
        title = new Label("Enter three numbers");
        result_label = new Label("Maximum Number");
        result_button = new Button("SUBMIT");
        title.setBounds(10,0,200,20);
        l1.setBounds(75,30,100,20);
        t1.setBounds(50,50,100,20);
        l2.setBounds(195,30,100,20);
        t2.setBounds(170,50,100,20);
        l3.setBounds(315,30,100,20);
        t3.setBounds(290,50,100,20);
        result_label.setBounds(170,80,110,20);
        result.setBounds(170,100,100,20);
        result_button.setBounds(180,130,80,30);
        add(t1);
        add(t2);
        add(t3);
        add(result_label);
        add(result);
        add(result_button);
        add(l1);
        add(l2);
        add(l3);
        add(title);
        result_button.addActionListener(this);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
//        throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
        float num1,num2,num3;
        String str;
        str = t1.getText();
        num1 = Float.parseFloat(str);
        str = t2.getText();
        num2 = Float.parseFloat(str);
        str = t3.getText();
        num3 = Float.parseFloat(str);
        if(num1>num2 && num1>num3)
            result.setText(num1+"");
        else if(num2>num1 && num2>num3)
            result.setText(num2+"");
        else
            result.setText(num3+"");
    }
}
```

OUTPUT:



```
Enter three numbers
Num 1: 100
Num 2: 150
Num 3: 200
Maximum Number: 200.0
Applet started.
```

RESULT:

The program is successfully executed and the output is verified.

PROGRAM NO: 37

AIM:

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

ALGORITHM:

```
Step.1: Start the program.

Step.2: Define a class 'CO5Q3' that extends Applet class and implements ActionListener interface.
Step.3: Using TextField class object, construct textfields to receive marks of 5 subjects from 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 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 a happy face if student secures above 50% or a sad face if otherwise

Step.9: Stop the program.
```

PROGRAM CODE:

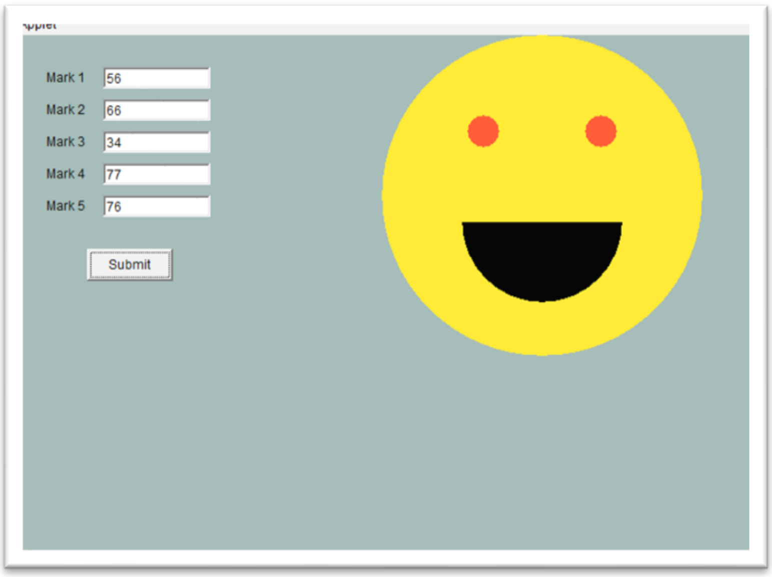
CO5Q3.java

```java
//Find the percentage of marks obtained by a student in 5 subjects. Display a happy face if
//he secures above 50% or a sad face if otherwise.
package javaprograms;

import java.applet.Applet;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;


public class CO5Q3 extends Applet implements ActionListener
{
    TextField t1,t2,t3,t4,t5;
```

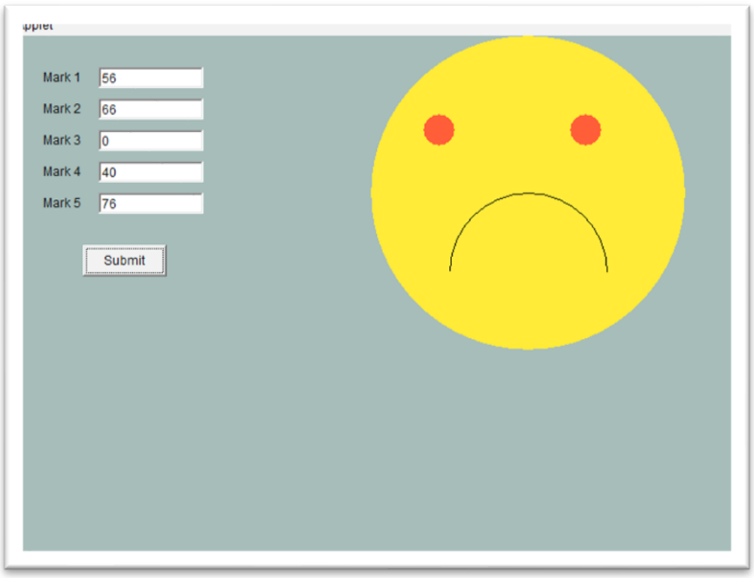
```
Button submit;
Label l1,l2,l3,l4,l5;
float num1,num2,num3,num4,num5,sum,percentage;
String str;
@Override
public void init()
{
    setLayout(null);
    setSize(700, 500);
    setBackground(Color.decode("#A4BAB7"));
    setForeground(Color.decode("#080F0F"));
    t1 = new TextField();
    t2 = new TextField();
    t3 = new TextField();
    t4 = new TextField();
    t5 = new TextField();
    l1 = new Label("Mark 1");
    l2 = new Label("Mark 2");
    l3 = new Label("Mark 3");
    l4 = new Label("Mark 4");
    l5 = new Label("Mark 5");
    submit = new Button("Submit");
    l1.setBounds(30,30,100,20);
    t1.setBounds(85,30,100,20);
    l2.setBounds(30,60,100,20);
    t2.setBounds(85,60,100,20);
    l3.setBounds(30,90,100,20);
    t3.setBounds(85,90,100,20);
    l4.setBounds(30,120,100,20);
    t4.setBounds(85,120,100,20);
    l5.setBounds(30,150,100,20);
    t5.setBounds(85,150,100,20);
    submit.setBounds(70, 200, 80, 30);
    add(t1);
    add(t2);
    add(t3);
    add(t4);
    add(t5);
    add(l1);
    add(l2);
    add(l3);
    add(l4);
    add(l5);
    add(submit);
    submit.addActionListener(this);
}
@Override
public void actionPerformed(ActionEvent e) {

    str = t1.getText();
    num1 = Float.parseFloat(str);
    str = t2.getText();
    num2 = Float.parseFloat(str);
    str = t3.getText();
    num3 = Float.parseFloat(str);
```

```
            str = t4.getText();
            num4 = Float.parseFloat(str);
            str = t5.getText();
            num5 = Float.parseFloat(str);
            sum = num1+num2+num3+num4+num5;
            percentage = (sum*100)/500;
            System.out.println(sum+" "+percentage);
            repaint();
        }
        @Override
        public void paint(Graphics g)
        {
            if(percentage > 50.0){
                    g.setColor(Color.decode("#FFE933"));
                 g.fillOval(345,0,300,300);
                 g.setColor(Color.decode("#FF5933"));
                 g.fillOval(425,75,30,30);
                 g.fillOval(535,75,30,30);
                 g.setColor(Color.black);
                 g.fillArc (420,100,150,150,0,-180);
            }
            else {
                g.setColor(Color.decode("#FFE933"));
                g.fillOval(345,0,300,300);
                g.setColor(Color.decode("#FF5933") );
                g.fillOval(395,75,30,30);
                g.fillOval(535,75,30,30);
                g.setColor(Color.black);
                g.drawArc(420,150,150,150,0,180);
            }
        }

    }
```

OUTPUT:



Mark 1 56
Mark 2 66
Mark 3 34
Mark 4 77
Mark 5 76



Mark 1 56
Mark 2 66
Mark 3 0
Mark 4 40
Mark 5 76

RESULT:

The program is successfully executed and the output is verified.

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 'CO5Q4 ' 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.

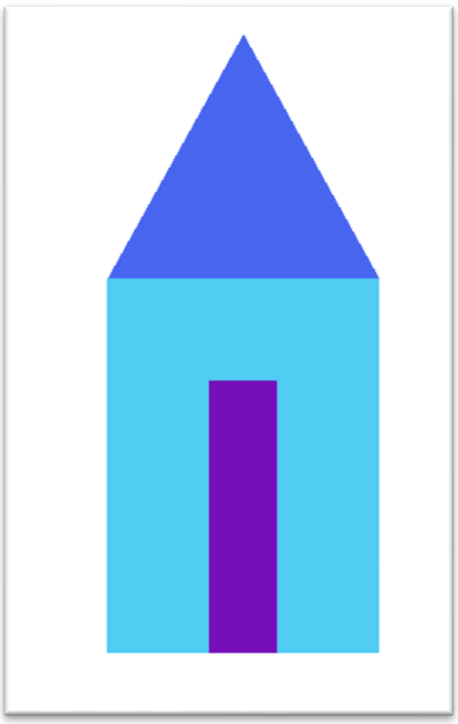
PROGRAM CODE:

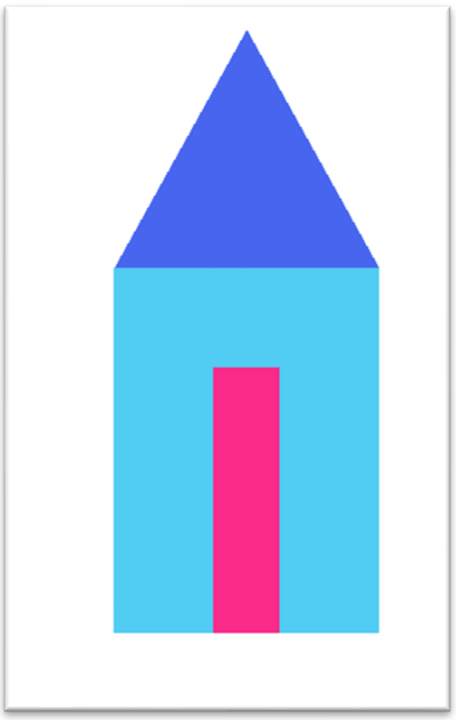
CO5Q4.java

```
//Using 2D graphics commands in an Applet, construct a house. On mouse click
event,
//change the color of the door from blue to red.
package javaprograms;

import java.applet.Applet;
import java.awt.*;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;

public class CO5Q4 extends Applet implements MouseListener
{
   int a, b;
            public void init() {
                        addMouseListener(this);
                        }
            public void paint (Graphics g) {

                        int x[]= {200,300,400};
                        int y[]= {200,20,200};

                        g.setColor(Color.decode("#4361EE"));
                        g.fillPolygon(x,y,3);
                        g.setColor(Color.decode("#4CC9F0"));
                        g.fillRect(200, 200, 200, 275);
                        g.setColor(Color.decode("#7209B7"));
```

```
                    g.fillRect(275, 275, 50, 200);

                    if(a>269 && a<321 && b>275 && b<470) {
          g.setColor(Color.decode("#F72585"));
          g.fillRect(275, 275, 50, 200);
        }

            }

    @Override
    public void mouseClicked(MouseEvent e) {

    }

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

    @Override
    public void mouseReleased(MouseEvent e) {

    }

    @Override
    public void mouseEntered(MouseEvent e) {

    }

    @Override
    public void mouseExited(MouseEvent e) {

    }

}
```

OUTPUT:





RESULT:

The program is successfully executed and the output is verified.

PROGRAM NO: 39

AIM:

To implement a simple calculator using AWT components.

ALGORITHM:

Step.1: Start the program.

Step.2: Define a class ‘CO5Q5 ‘ 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.8: Stop the program.

PROGRAM CODE:

CO5Q5.
java
	package javaprograms;
	//Implement a simple calculator using AWT components.
	
	import java.applet.Applet;
	import java.awt.*;
	import java.awt.event.ActionEvent;
	import java.awt.event.ActionListener;
	
	public class CO5Q5 extends Applet implements ActionListener
	{
	    TextField t1;

```
    Button
b0,b1,b2,b3,b4,b5,b6,b7,b8,b9,button_add,button_sub,button_mul,button_div,b
utton_equal,button_clear,button_dot;
    Label l1;
    @Override
    public void init()
    {
        setLayout(null);
        l1 = new Label("Calculator");
        t1 = new TextField();
        b0 = new Button("0");
        b1 = new Button("1");
        b2 = new Button("2");
        b3 = new Button("3");
        b4 = new Button("4");
        b5 = new Button("5");
        b6 = new Button("6");
        b7 = new Button("7");
        b8 = new Button("8");
        b9 = new Button("9");
        button_add = new Button("+");
        button_sub = new Button("-");
        button_mul = new Button("x");
        button_div = new Button("/");
        button_equal = new Button("=");
        button_clear = new Button("AC");
        button_dot = new Button(".");
        l1.setBounds(50,0,2000,50);
        Font myFont = new Font("Serif",Font.BOLD,50);
        l1.setFont(myFont);
        t1.setBounds(50, 100, 240, 30);
        button_clear.setBounds(290, 100, 80, 30);
        b1.setBounds(50,130,80,30);
        b2.setBounds(130,130,80,30);
        b3.setBounds(210,130,80,30);
        button_add.setBounds(290,130,80,30);
        b4.setBounds(50,160,80,30);
        b5.setBounds(130,160,80,30);
        b6.setBounds(210,160,80,30);
        button_sub.setBounds(290,160,80,30);
        b7.setBounds(50,190,80,30);
        b8.setBounds(130,190,80,30);
        b9.setBounds(210,190,80,30);
        button_mul.setBounds(290,190,80,30);
        button_dot.setBounds(50,220,80,30);
        b0.setBounds(130,220,80,30);
        button_equal.setBounds(210,220,80,30);
        button_div.setBounds(290,220,80,30);
        button_equal.setForeground(Color.decode("#E1F4F1"));
        button_equal.setBackground(Color.decode("#25BA9F"));
```

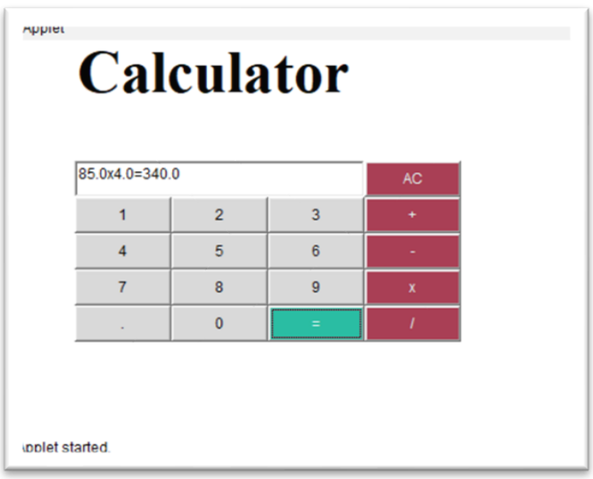
```
b0.setBackground(Color.decode("#D6D6D6"));
b1.setBackground(Color.decode("#D6D6D6"));
b2.setBackground(Color.decode("#D6D6D6"));
b3.setBackground(Color.decode("#D6D6D6"));
b4.setBackground(Color.decode("#D6D6D6"));
b5.setBackground(Color.decode("#D6D6D6"));
b6.setBackground(Color.decode("#D6D6D6"));
b7.setBackground(Color.decode("#D6D6D6"));
b8.setBackground(Color.decode("#D6D6D6"));
b9.setBackground(Color.decode("#D6D6D6"));
button_dot.setBackground(Color.decode("#D6D6D6"));
button_clear.setBackground(Color.decode("#A63A50"));
button_add.setBackground(Color.decode("#A63A50"));
button_sub.setBackground(Color.decode("#A63A50"));
button_mul.setBackground(Color.decode("#A63A50"));
button_div.setBackground(Color.decode("#A63A50"));
button_add.setForeground(Color.decode("#E1F4F1"));
button_sub.setForeground(Color.decode("#E1F4F1"));
button_mul.setForeground(Color.decode("#E1F4F1"));
button_div.setForeground(Color.decode("#E1F4F1"));
button_clear.setForeground(Color.decode("#E1F4F1"));
add(l1);
add(t1);
add(b1);
add(b2);
add(b3);
add(b4);
add(b5);
add(b6);
add(b7);
add(b8);
add(b9);
add(b0);
add(button_dot);
add(button_equal);
add(button_add);
add(button_clear);
add(button_sub);
add(button_mul);
add(button_div);
b0.addActionListener(this);
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
b5.addActionListener(this);
b6.addActionListener(this);
b7.addActionListener(this);
b8.addActionListener(this);
b9.addActionListener(this);
```

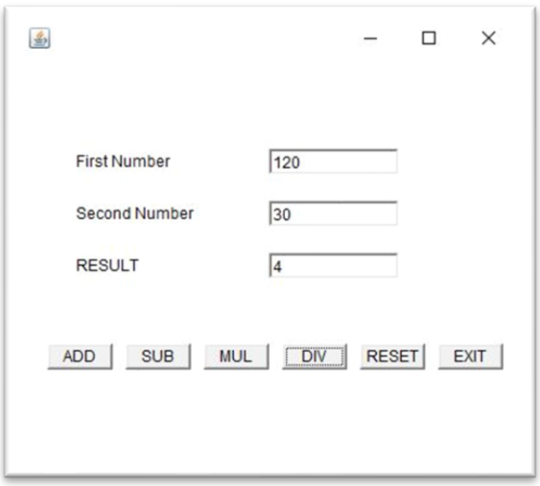
```
        button_add.addActionListener(this);
        button_sub.addActionListener(this);
        button_mul.addActionListener(this);
        button_div.addActionListener(this);
        button_dot.addActionListener(this);
        button_clear.addActionListener(this);
        button_equal.addActionListener(this);
    }
    int op = 5;
    boolean flag=false;
    String str1,str2,str_result;
    float num1=0,num2=0,result;
    @Override
    public void actionPerformed(ActionEvent e) {
//          throw new UnsupportedOperationException("Not supported yet."); //To
change body of generated methods, choose Tools | Templates.
        if(e.getSource()==b0)
            if(flag!=true)
            {
                str1 = t1.getText();
                t1.setText(str1+"0");
            }
            else
            {
                str2 = t1.getText();
                t1.setText(str2+"0");
            }
        if(e.getSource()==b1)
            if(flag!=true)
            {
                str1 = t1.getText();
                t1.setText(str1+"1");
            }
            else
            {
                str2 = t1.getText();
                t1.setText(str2+"1");
            }
        if(e.getSource()==b2)
            if(flag!=true)
            {
                str1 = t1.getText();
                t1.setText(str1+"2");
            }
            else
            {
                str2 = t1.getText();
                t1.setText(str2+"2");
            }
        if(e.getSource()==b3)
    if(flag!=true)
    {
        str1 = t1.getText();
        t1.setText(str1+"3");
    }
    else
    {
        str2 = t1.getText();
        t1.setText(str2+"3");
    }
if(e.getSource()==b4)
    if(flag!=true)
    {
        str1 = t1.getText();
        t1.setText(str1+"4");
    }
    else
    {
        str2 = t1.getText();
        t1.setText(str2+"4");
    }
if(e.getSource()==b5)
    if(flag!=true)
    {
        str1 = t1.getText();
        t1.setText(str1+"5");
    }
    else
    {
        str2 = t1.getText();
        t1.setText(str2+"5");
    }
if(e.getSource()==b6)
    if(flag!=true)
    {
        str1 = t1.getText();
        t1.setText(str1+"6");
    }
    else
    {
        str2 = t1.getText();
        t1.setText(str2+"6");
    }
if(e.getSource()==b7)
    if(flag!=true)
    {
        str1 = t1.getText();
        t1.setText(str1+"7");
    }
    else
```

```
            {
                str2 = t1.getText();
                t1.setText(str2+"7");
            }
        if(e.getSource()==b8)
            if(flag!=true)
            {
                str1 = t1.getText();
                t1.setText(str1+"8");
            }
            else
            {
                str2 = t1.getText();
                t1.setText(str2+"8");
            }
        if(e.getSource()==b9)
            if(flag!=true)
            {
                str1 = t1.getText();
                t1.setText(str1+"9");
            }
            else
            {
                str2 = t1.getText();
                t1.setText(str2+"9");
            }
        if(e.getSource()==button_dot)
            if(flag!=true)
            {
                str1 = t1.getText();
                t1.setText(str1+".");
            }
            else
            {
                str2 = t1.getText();
                t1.setText(str2+".");
            }
        if(e.getSource()==button_clear)
        {
            t1.setText("");
        }
        if(e.getSource()==button_add)
        {
            flag=true;
            op=1;
            num1=Float.parseFloat(t1.getText());
            t1.setText("");
        }
        if(e.getSource()==button_sub)
        {
          flag=true;
          op=2;
          num1=Float.parseFloat(t1.getText());
          t1.setText("");
      }
      if(e.getSource()==button_mul)
      {
          flag=true;
          op=3;
          num1=Float.parseFloat(t1.getText());
          t1.setText("");
      }
      if(e.getSource()==button_div)
      {
          flag=true;
          op=4;
          num1=Float.parseFloat(t1.getText());
          t1.setText("");
      }
      if(e.getSource()==button_equal)
      {
          num2 = Float.parseFloat(t1.getText());
          switch(op)
          {
              case 1: result = num1 + num2;
                      str_result = String.valueOf(num1);
                      str_result += "+";
                      str_result += String.valueOf(num2);
                      str_result += "=";
                      str_result += String.valueOf(result);
                      t1.setText(str_result);
                      break;
              case 2: result = num1 - num2;
                      t1.setText(String.valueOf(result));
                      str_result = String.valueOf(num1);
                      str_result += "-";
                      str_result += String.valueOf(num2);
                      str_result += "=";
                      str_result += String.valueOf(result);
                      t1.setText(str_result);
                      break;
              case 3: result = num1 * num2;
                      str_result = String.valueOf(num1);
                      str_result += "x";
                      str_result += String.valueOf(num2);
                      str_result += "=";
                      str_result += String.valueOf(result);
                      t1.setText(str_result);
                      break;
              case 4: result = num1 / num2;
```

```
                              str_result = String.valueOf(num1);
                              str_result += "/";
                              str_result += String.valueOf(num2);
                              str_result += "=";
                              str_result += String.valueOf(result);
                              t1.setText(str_result);
                              break;
                    default:t1.setText("Error");
                }
            }
        }
    }
```

OUTPUT:





RESULT:

The program is successfully executed and the output is verified.

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 an interface 'CO5Q7' 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 from the menu and repaint accordingly.

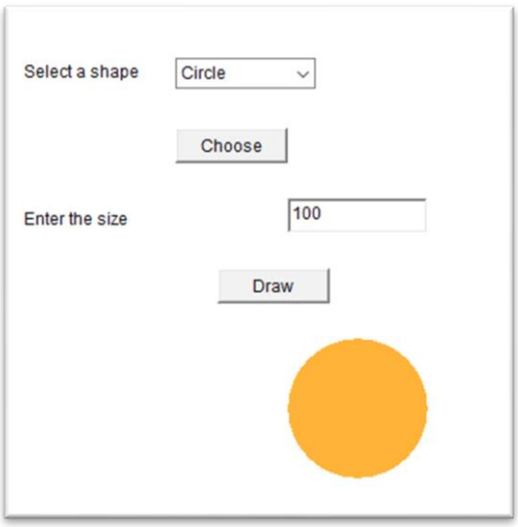
Step.6: Stop the program.

PROGRAM CODE:

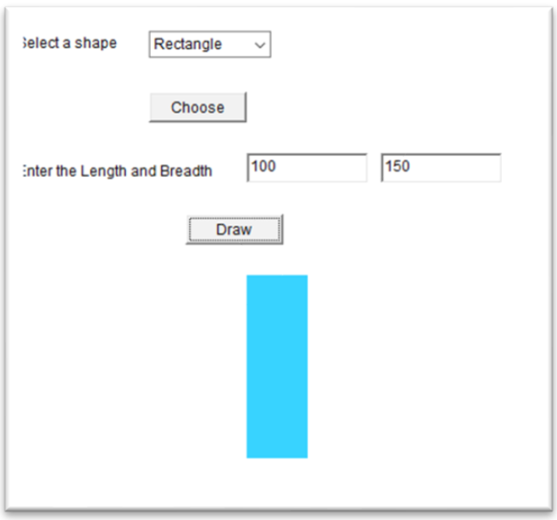
CO5Q7.java

```
//Develop a program to handle all mouse events and window events
package javaprograms;

import java.applet.Applet;
import java.awt.Graphics;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.awt.event.MouseMotionListener;

public class CO5Q7 extends Applet implements
MouseListener,MouseMotionListener
{
    String s = "";
    int x=0,y=0;
    @Override
    public void init()
    {
        addMouseListener(this);
        addMouseMotionListener(this);
    }
    @Override
    public void mouseClicked(MouseEvent e) {
            s = "Mouse Clicked";
            repaint();
        }
        @Override
        public void mousePressed(MouseEvent e) {
            s = "Mouse Pressed";
            repaint();
        }
        @Override
        public void mouseReleased(MouseEvent e) {
            s = "Mouse Released";
            repaint();
        }
        @Override
        public void mouseEntered(MouseEvent e) {
            s = "Mouse Entered";
            repaint();
        }
        @Override
        public void mouseExited(MouseEvent e) {
            s = "Mouse Exited";
            repaint();
        }
        @Override
        public void mouseDragged(MouseEvent e) {
            s = "Mouse Dragged";
            x = e.getX();
            y = e.getY();
            showStatus("Mouse Position: X="+x+" Y="+y);
            repaint();
        }
        @Override
        public void mouseMoved(MouseEvent e) {
            s = "Mouse Moved";
            x = e.getX();
            y = e.getY();
            showStatus("Mouse Position: X="+x+" Y="+y);
            repaint();
        }
        @Override
        public void paint(Graphics g)
        {
            g.drawString("Handling Mouse Events",30,20);
            g.drawString(s,60,40);
        }


    }
```

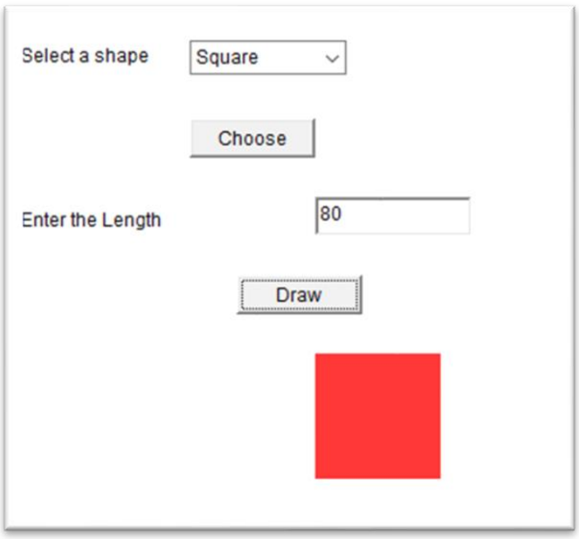
OUTPUT:


Select a shape: Circle

Enter the size: 100


Select a shape: Rectangle

Enter the Length and Breadth: 100 150


Select a shape: Square

Enter the Length: 80

RESULT:

The program is successfully executed and the output is verified.

PROGRAM: 41

AIM : Develop a program to handle all mouse events and window events

ALGORITHM :

Step.1: Start

Step.2: Define a class 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.

Step.7: Display the appropriate message in the output.

Step.8: Stop

PROGRAM CODE :

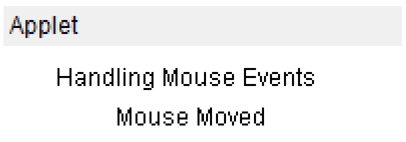
CO5Q7.java

```
//Develop a program to handle all mouse events and window events
package javaprograms;

import java.applet.Applet;
import java.awt.Graphics;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.awt.event.MouseMotionListener;

public class CO5Q7 extends Applet implements
MouseListener,MouseMotionListener
{
   String s = "";
   int x=0,y=0;
   @Override
   public void init()
   {
      addMouseListener(this);
      addMouseMotionListener(this);
   }
   @Override
   public void mouseClicked(MouseEvent e) {
      s = "Mouse Clicked";
      repaint();
   }
   @Override
   public void mousePressed(MouseEvent e) {
      s = "Mouse Pressed";
      repaint();
   }
   @Override
   public void mouseReleased(MouseEvent e) {
      s = "Mouse Released";
      repaint();
   }
   @Override
   public void mouseEntered(MouseEvent e) {
      s = "Mouse Entered";
      repaint();
   }
   @Override
   public void mouseExited(MouseEvent e) {
      s = "Mouse Exited";
      repaint();
```

```java
    }
    @Override
    public void mouseDragged(MouseEvent e) {
        s = "Mouse Dragged";
        x = e.getX();
        y = e.getY();
        showStatus("Mouse Position: X="+x+" Y="+y);
        repaint();
    }
    @Override
    public void mouseMoved(MouseEvent e) {
        s = "Mouse Moved";
        x = e.getX();
        y = e.getY();
        showStatus("Mouse Position: X="+x+" Y="+y);
        repaint();
    }
    @Override
    public void paint(Graphics g)
    {
        g.drawString("Handling Mouse Events",30,20);
        g.drawString(s,60,40);
    }


}
```

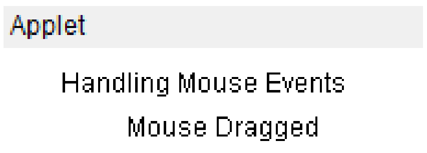
<u>RESULT :</u> The above program is successfully executed and obtained the output

OUTPUT :


Applet

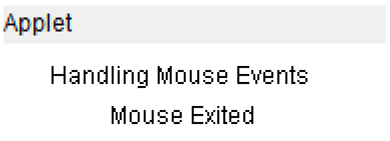
Handling Mouse Events

Mouse Moved


Applet

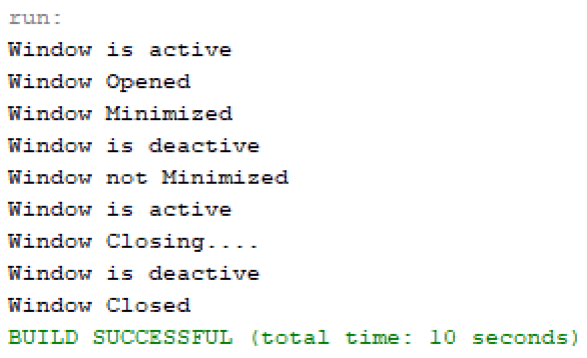
Handling Mouse Events

Mouse Dragged


Applet

Handling Mouse Events

Mouse Exited


```
run:
Window is active
Window Opened
Window Minimized
Window is deactive
Window not Minimized
Window is active
Window Closing....
Window is deactive
Window Closed
BUILD SUCCESSFUL (total time: 10 seconds)
```

PROGRAM NO: 42

AIM:

To develop a program to handle Key events.

ALGORITHM:

Step.1: Start the program.

Step.2: Define a class CO5Q8 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.

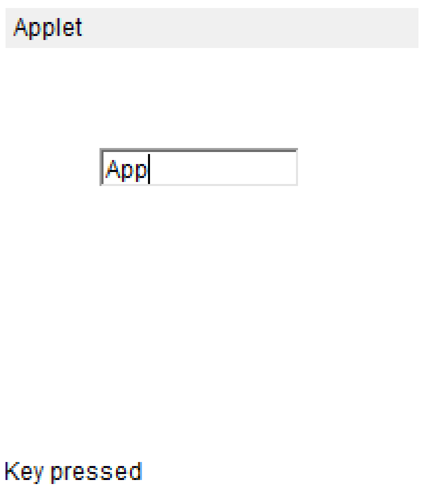
Step.5: Stop the program.

PROGRAM CODE:

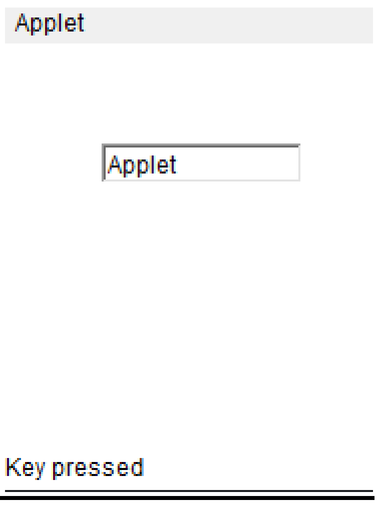
CO5Q8.java

```java
//Develop a program to handle Key events
package javaprograms;

import java.applet.Applet;
import java.awt.Graphics;
import java.awt.TextField;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;

public class CO5Q8 extends Applet implements KeyListener
{
    String msg="";
    TextField t1;
    @Override
    public void init()
    {
        setLayout(null);
        addKeyListener(this);
        t1 = new TextField();
        t1.setBounds(50,50,100,20);
        t1.addKeyListener(this);
        add(t1);
    }
    @Override
    public void keyTyped(KeyEvent e) {
        msg+= e.getKeyChar();
        repaint();
    }

    @Override
    public void keyPressed(KeyEvent e) {
        showStatus("Key pressed");
    }

    @Override
    public void keyReleased(KeyEvent e) {
        showStatus("Key released");
    }
    @Override
    public void paint(Graphics g)
    {

    }
}
```

OUTPUT:



```
Applet

App

Key pressed
```



```
Applet

Applet

Key pressed
```

RESULT:

The program is successfully executed and the output is verified.

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 ‘CO6Q1’ that implements FilenameFilter interface.

Step.3: Create an object of 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.

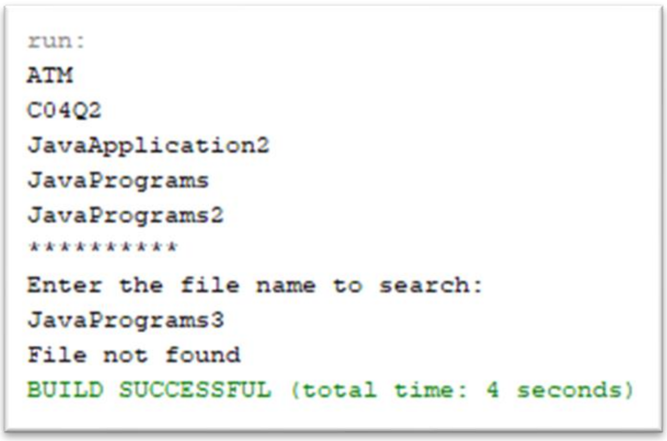
Step.8: Stop the program.

PROGRAM CODE:

CO6Q1.java	
	```
	//Program to list the sub directories and files in a given directory and also
	search for a file
	//name.
	package javaprograms;

	import java.io.File;
	import java.util.Scanner;

	public class CO6Q1 {
	    public static void main(String[] args) {
	        int flag=0;
	        File f = new File("../");
	        Scanner sc = new Scanner(System.in);
	        String name,filename;
	```

	```
	        String[] filelist = f.list();
	        try{
	        for(String str: filelist)
	           System.out.println(str);
	        System.out.println("**********");
	        System.out.println("Enter the file name to search:");
	        name = sc.nextLine();
	        for(int i =0; i<f.length();i++){
	           filename = filelist[i];
	           if(filename.equals(name)){
	              flag=1;
	              break;
	           }
	           else
	              flag=0;
	        }}
	        catch(Exception e)
	        {
	        if(flag==1)
	           System.out.println("File found");
	        else
	           System.out.println("File not found");
	        }
	     }
	}
	```

OUTPUT:



```
run:
ATM
C04Q2
JavaApplication2
JavaPrograms
JavaPrograms2
**********
Enter the file name to search:
JavaPrograms3
File not found
BUILD SUCCESSFUL (total time: 4 seconds)
```

RESULT:

The program is successfully executed and the output is verified.

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 ‘CO6Q2’.

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

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

Step.5: Create and use an object of 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.

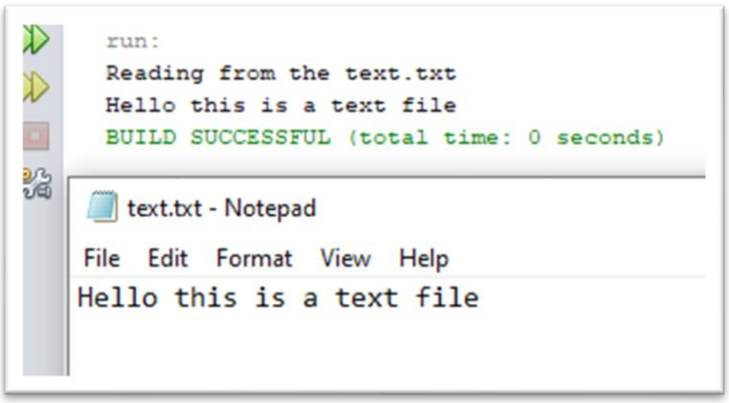
PROGRAM CODE:

CO6Q2.java

```
//Write a program to write to a file, then read from the file and display the
contents on the
//console.
package javaprograms;
import java.io.*;

public class CO6Q2 {
    public static void main(String[] args) {
        try
        {
            String str;
            FileWriter fw = new FileWriter("text.txt",true);
            fw.write("Hello this is a text file");
            fw.close();
            FileReader fr = new FileReader("text.txt");
            BufferedReader br = new BufferedReader(fr);
            System.out.println("Reading from the text.txt");
            while((str=br.readLine())!=null)
                System.out.println(str);
```

	```
	        }
	        catch(Exception e)
	        {

	        }
	    }
	}
	```

OUTPUT:



```
run:
Reading from the text.txt
Hello this is a text file
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
Hello this is a text file
```

RESULT:

The program is successfully executed and the output is verified.

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 ‘CO6Q3’.

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

Step.4: Create and use an object of 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.

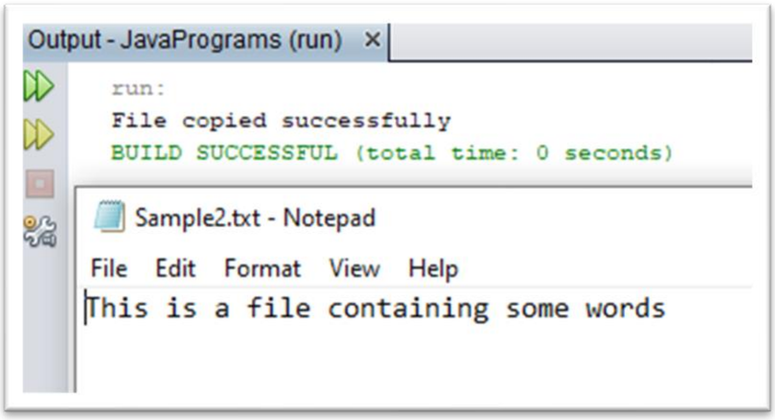
Step.7: Stop the program.

PROGRAM CODE:

CO6Q3.java	
	```
	//Write a program to copy one file to another.
	package javaprograms;

	import java.io.*;
	import java.util.*;

	public class CO6Q3 {
	    public static void main(String[] args) {
	        try {
	            String str;
	            FileReader fr = new FileReader("Sample1.txt");
	            BufferedReader br = new BufferedReader(fr);
	            FileWriter fw = new FileWriter("Sample2.txt",true);
	            BufferedWriter bw = new BufferedWriter(fw);
	            while((str=br.readLine())!=null)
	            {
	```

```
                fw.write(str);
                fw.flush();
            }
            System.out.println("File copied successfully");
        }
        catch (Exception ex) {
            System.out.println(ex);
        }
    }
}
```

OUTPUT:



```
run:
File copied successfully
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
This is a file containing some words
```

RESULT:

The program is successfully executed and the output is verified.

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 'CO6Q4'.

Step.3: Create an object of 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 of the FileWriter class, write those integers into the file.

Step.7: Using objects of 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);
      }
```

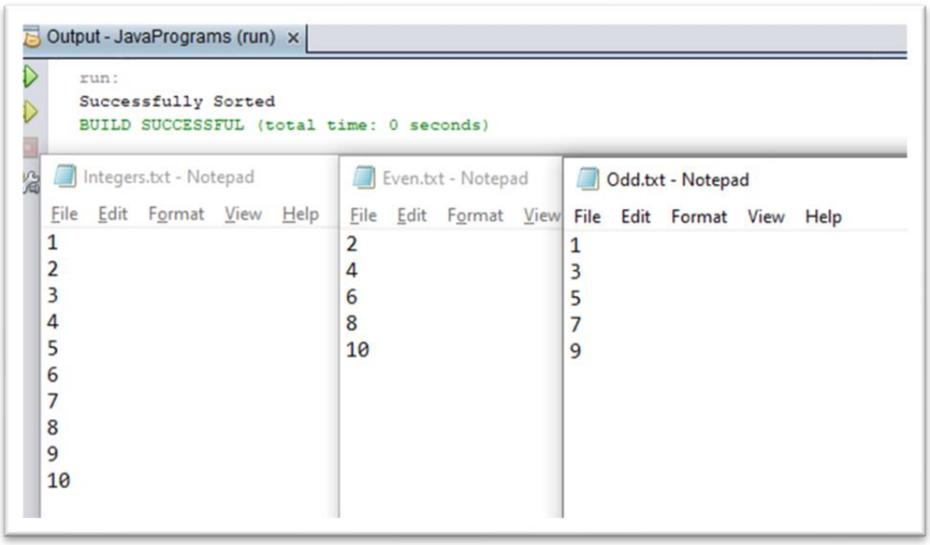
Step.8: Stop the program.

PROGRAM CODE:

CO6Q4.java	
	```
	//Write a program that reads from a file having integers. Copy even numbers and odd
	//numbers to separate files.
	package javaprograms;

	import java.io.*;
	public class CO6Q4 {
	    public static void main(String[] args) {
	        try {
	            String str;
	            int n;
	```

```
            FileReader fr = new FileReader("Integers.txt");
            BufferedReader br = new BufferedReader(fr);
            FileWriter fileEven = new FileWriter("Even.txt",true);
            FileWriter fileOdd = new FileWriter("Odd.txt",true);
            BufferedWriter bwEven = new BufferedWriter(fileEven);
            BufferedWriter bwOdd = new BufferedWriter(fileOdd);
            while((str=br.readLine())!=null){
                n = Integer.parseInt(str);
                if(n%2==0){
                    bwEven.write(String.valueOf(n));
                    bwEven.write("\n");
                    bwEven.flush();
                }
                else{
                    bwOdd.write(String.valueOf(n));
                    bwOdd.write("\n");
                    bwOdd.flush();
                }
            }
            System.out.println("Successfully Sorted");
        } catch (Exception e) {
        }}}
```

OUTPUT:



Output - JavaPrograms (run)

```
run:
Successfully Sorted
BUILD SUCCESSFUL (total time: 0 seconds)
```

Integers.txt - Notepad

```
1
2
3
4
5
6
7
8
9
10
```

Even.txt - Notepad

```
2
4
6
8
10
```

Odd.txt - Notepad

```
1
3
5
7
9
```

RESULT:

The program is successfully executed and the output is verified.

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.
```

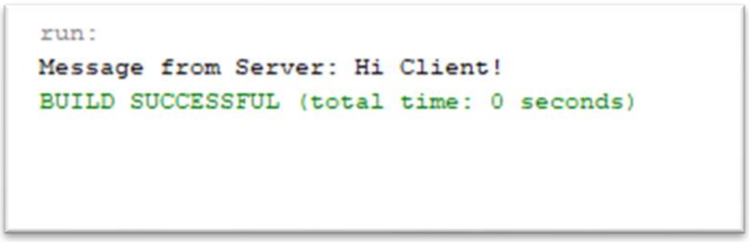
PROGRAM CODE:

CO6Q5Client.java	
	```
	//Client server communication using Socket – TCP/IP
	package javaprograms;
	import java.net.*;
	import java.io.*;

	public class CO6Q5Client {
	    public static void main(String args[]) throws Exception{
	    try {
	            Socket s = new Socket ("localhost", 5555);
	            PrintWriter pw = new PrintWriter(s.getOutputStream(), true);
	            pw.println("Hello Server!");
	        InputStreamReader isr = new InputStreamReader(s.getInputStream());
	            BufferedReader br = new BufferedReader(isr);
	            String str= br.readLine();
	            System.out.println("Message from Server: "+str);
	            pw.close();
	            s.close();
	            }
	            catch(Exception e) {
	           System.out.println("An error occured..." +e);
	            }}}
	```

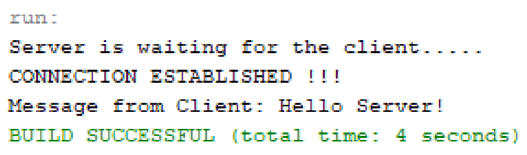
CO6Q5Server.java	
	```
	//Client server communication using Socket – TCP/IP
	package javaprograms;
	```

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

	public class CO6Q5Server {
	    public static void main(String[] args) throws Exception {
	                try {
	            ServerSocket ss = new ServerSocket(5555);
	            System.out.println("Server is waiting for the client.....");
	            Socket s = ss.accept();
	            System.out.println("CONNECTION ESTABLISHED !!!");
	            InputStreamReader isr = new InputStreamReader(s.getInputStream());
	            BufferedReader br = new BufferedReader(isr);
	            String str = br.readLine();
	            System.out.println("Message from Client: "+str);
	            PrintWriter pw = new PrintWriter(s.getOutputStream(), true);
	            pw.println("Hi Client!");
	            pw.close();
	            }
	                catch(Exception e) {
	                        System.out.println("An error occured.."+e);
	                }}}
	```

OUTPUT:



```
run:
Message from Server: Hi Client!
BUILD SUCCESSFUL (total time: 0 seconds)
```



```
run:
Server is waiting for the client.....
CONNECTION ESTABLISHED !!!
Message from Client: Hello Server!
BUILD SUCCESSFUL (total time: 4 seconds)
```

RESULT:

The program is successfully executed and the output is verified.

PROGRAM NO: 48

AIM:

To implement 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:

CO6Q6Client.java	//Client Server communication using DatagramSocket - UDP
	package javaprograms;
	import java.io.IOException;
	import java.net.InetAddress;
	import java.net.SocketException;
	import java.net.DatagramSocket;
	import java.net.DatagramPacket;

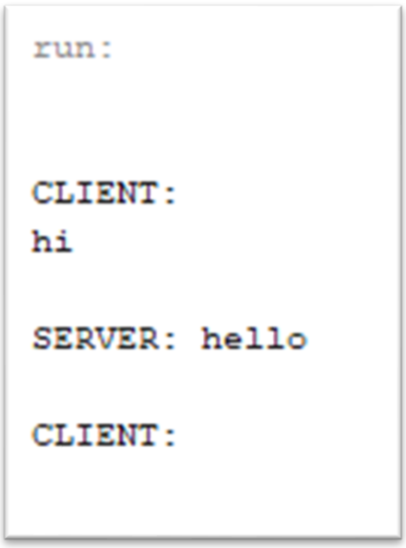
```
import java.util.Scanner;

public class CO6Q6Client {
    public static void main(String[] args) throws SocketException, IOException {
            InetAddress IP = InetAddress.getByName("localhost");
            DatagramSocket csocket = new DatagramSocket();
                    while(true) {
                            byte[] sendbuffer = new byte[1024];
                            byte[] receivebuffer = new byte[1024];
                            System.out.println("\n\nCLIENT: ");
                            Scanner sc = new Scanner(System.in);
                            String clientData = sc.nextLine();
                            sendbuffer = clientData.getBytes();

                            DatagramPacket sndpkt = new DatagramPacket(sendbuffer,
sendbuffer.length, IP, 9876);
                            csocket.send(sndpkt);;

                            if(clientData.equalsIgnoreCase("Bye")) {
                                    System.out.println("Connection dropped by Client. .!");
                                    break;
                            }
                            DatagramPacket rcvpkt = new DatagramPacket(receivebuffer,
receivebuffer.length);
                            csocket.receive(rcvpkt);
                            String serverData = new String(rcvpkt.getData());
                            System.out.print("\nSERVER: "+ serverData);
                    }
                    csocket.close();
            }
}
```

CO6Q6Server.java

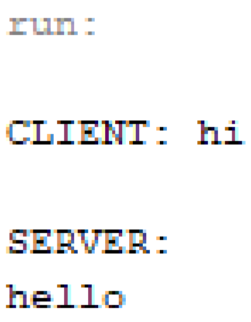
```
package javaprograms;
import java.io.IOException;
import java.net.InetAddress;
import java.net.SocketException;
import java.net.DatagramSocket;
import java.net.DatagramPacket;
import java.util.Scanner;

public class CO6Q6Server {
    public static void main(String[] args) throws SocketException, IOException {
                    DatagramSocket sSocket = new DatagramSocket(9876);
                    while(true) {
                            byte[] sendbuffer = new byte[1024];
                            byte[] receivebuffer = new byte[1024];
                            DatagramPacket rcvdpkt = new DatagramPacket(receivebuffer,
receivebuffer.length);
                            sSocket.receive(rcvdpkt);
                            InetAddress IP = rcvdpkt.getAddress();
                            int portNo = rcvdpkt.getPort();
                            String clData = new String(rcvdpkt.getData());
                            System.out.println("\nCLIENT: " + clData);
                            System.out.println("\nSERVER: ");
                            Scanner sc = new Scanner(System.in);
```

	```
	String serData = sc.nextLine();
	sendbuffer = serData.getBytes();
	DatagramPacket sendpkt = new DatagramPacket(sendbuffer, sendbuffer.length, IP, portNo);
	sSocket.send(sendpkt);

	if(serData.equalsIgnoreCase("Bye")) {
	    System.out.println("Connection dropped by Server. . !");
	    break;
	}
	}
	sSocket.close();
	}
	}
	```

OUTPUT:



```
run:


CLIENT:
hi

SERVER: hello

CLIENT:
```



```
run:

CLIENT: hi

SERVER:
hello
```

RESULT:

The program is successfully executed and the output is verified.