Exp No.: 1
Date:

Simple Java Programs

1. Write a java program to find the largest of three numbers.

AIM:

To write a JAVA program to find the largest of three numbers.

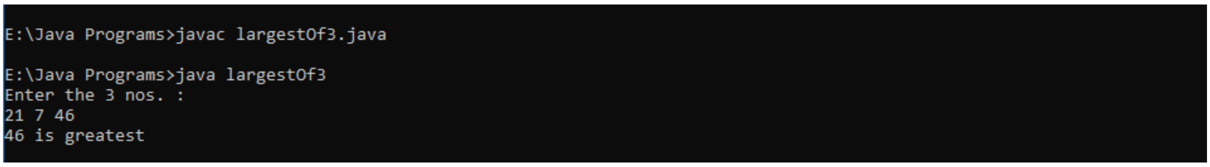
ALGORITHM:

```
Step 1: Start.
Step 2: Get 3 numbers from the user, namely, ‘a’, ’b’, ’c’.
Step 3: If ‘a’ is greater than both ‘b’ and ‘c’, print ‘a’ as the largest.
Step 4: Else if, ‘b’ is greater than both ‘a’ and ‘c’, print ‘b’ as the largest.
Step 5: Else, print ‘c’ as the largest.
Step 6: End.
```

PROGRAM:

```java
import java.util.Scanner;
public class largestOf3
{
    public static void main(String[] args)
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the 3 nos. : ");
        int a=s.nextInt();
        int b=s.nextInt();
        int c=s.nextInt();
        if(a>b&&a>c)
            System.out.printf("%d is greatest\n",a);
        else if(b>a&&b>c)
            System.out.printf("%d is greatest\n",b);
        else
            System.out.printf("%d is greatest\n",c);
        s.close();
    }
}
```

OUTPUT:



```
E:\Java Programs>javac largestOf3.java

E:\Java Programs>java largestOf3
Enter the 3 nos. :
21 7 46
46 is greatest
```

RESULT:

The java program to find the largest number of 3 numbers was implemented and executed successfully.

2. Write a java program to print the n^{th} value in the Fibonacci sequence.

AIM:

To write a JAVA program to print the n^{th} value in the Fibonacci sequence.

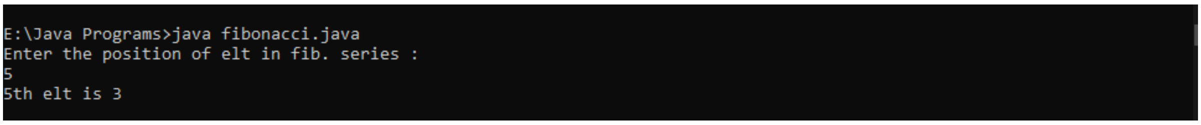
ALGORITHM:

Step 1: Start.
Step 2: Import required packages.
Step 3: Get the position of the element in the sequence from the user, namely, n.
Step 4: If n is 1, print 0.
Step 5: If n is 2, print 1.
Step 6: Else, calculate the element and print it.
Step 7: End.

PROGRAM:

```java
import java.util.Scanner;
public class fibonacci
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the position of elt in fib. series :");
        int n=s.nextInt();
        if(n==1)
            System.out.println("1st elt is 0");
        else if(n==2)
            System.out.println("2nd elt is 1");
        else
        {
            int f=0,sec=1,t=0,i=3;
            while(i<=n)
            {
                t=f+sec;
                f=sec;
                sec=t;
                i++;
            }
            System.out.println(n+"th elt is "+t);
        }
        s.close();
    }
}
```

OUTPUT:



```
E:\Java Programs>java fibonacci.java
Enter the position of elt in fib. series :
5
5th elt is 3
```

RESULT:

The JAVA program to print the nth value in the Fibonacci sequence was implemented and executed successfully.

3. Write a java program to check the number is prime or not.

AIM:

To write a JAVA program to check the number is prime or not.

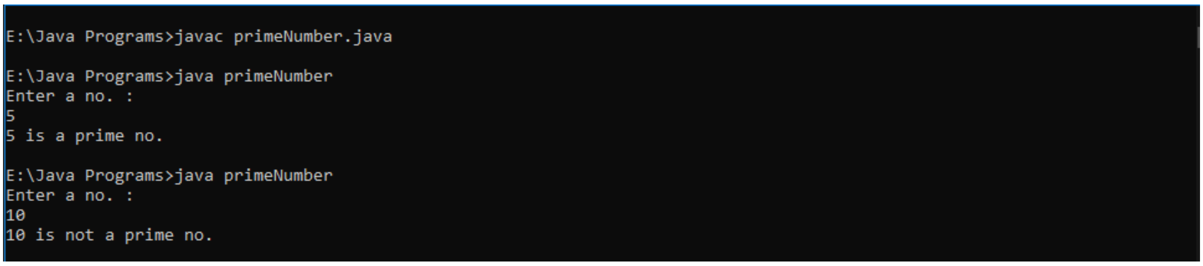
ALGORITHM:

Step 1: Start.
Step 2: Import required packages.
Step 3: Get the number to be checked from the user, n.
Step 4: Determine if the number is prime or not and print the same.
Step 5: End.

PROGRAM:

```
import java.util.Scanner;
public class primeNumber
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter a no. :");
        int n=s.nextInt();
        if(n==0||n==1)
            System.out.println(n+" is not a prime no.");
        else
        {
            int i=2,flag=0;
            for(;i<=n/2;i++)
            {
                if(n%i==0)
                {
                    flag=1;
                    break;
                }
            }
            if(flag==0)
                System.out.println(n+" is a prime no.");
            else
                System.out.println(n+" is not a prime no.");
        }

        s.close();
    }
}
```

OUTPUT:



```
E:\Java Programs>javac primeNumber.java

E:\Java Programs>java primeNumber
Enter a no. :
5
5 is a prime no.

E:\Java Programs>java primeNumber
Enter a no. :
10
10 is not a prime no.
```

RESULT:

The JAVA program to check the number is prime or not was implemented and executed successfully.

4. Write a java program to reverse a number using for, while, and recursion.

AIM:

To write a JAVA program to reverse a number using for, while, and recursion.

ALGORITHM:

Step 1: Start.
Step 2: Import required packages.
Step 3: Get the number to reversed from the user.
Step 4: The reverse of number is displayed by printing the unit digit and performing floor division, until the number is 0.
Step 5: End.

PROGRAM:

a) while:

```
import java.util.Scanner;
public class reverse1
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the no. :");
        int n=s.nextInt();
        int t,rev=0;
        t=n;
        while(t>0)
        {
            rev*=10;
            rev+=(t%10);
            t/=10;
        }
        System.out.println("Reverse of "+n+" is "+rev);
        s.close();
    }
}
```

b) for:

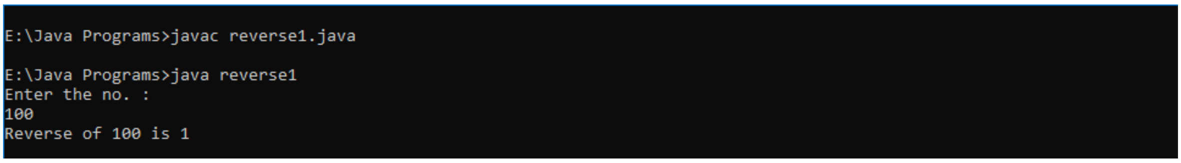
```
import java.util.Scanner;
public class reverse2
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the no. :");
        int n=s.nextInt();
        int rev=0;
        for(int t=n;t>0;t/=10)
        {
            rev*=10;
            rev+=(t%10);
        }
        System.out.println("Reverse of "+n+" is "+rev);
        s.close();
    }
}
```

c) recursion:

```
import java.util.Scanner;
public class reverse3
{
    static void revIt(int n)
    {
        if(n<10)
        {
            System.out.println(n);
            return;
        }
        else{
            System.out.print(n%10);
            revIt(n/10);
        }
    }
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the no. :");
        int n=s.nextInt();
        System.out.println("Reverse of "+n+" is :");
        revIt(n);
        s.close();
    }
}
```

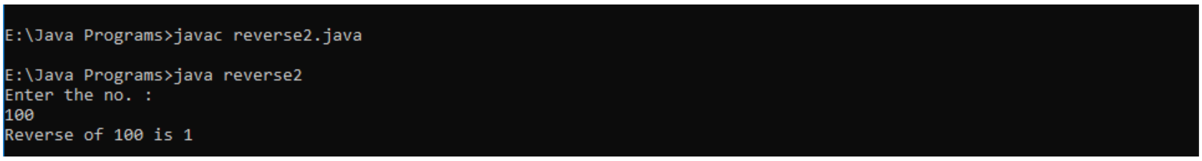
__**OUTPUT:**__

a) while:



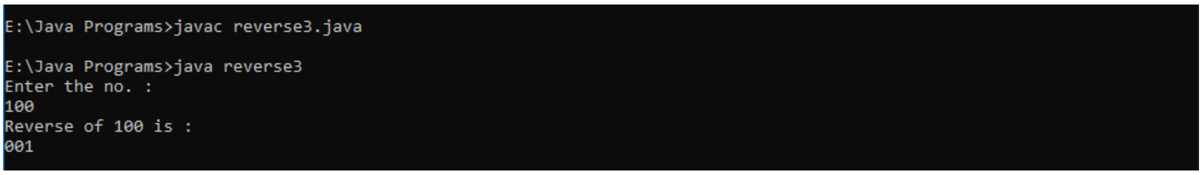
```
E:\Java Programs>javac reverse1.java

E:\Java Programs>java reverse1
Enter the no. :
100
Reverse of 100 is 1
```

b) for:



```
E:\Java Programs>javac reverse2.java

E:\Java Programs>java reverse2
Enter the no. :
100
Reverse of 100 is 1
```

c) recursion:



```
E:\Java Programs>javac reverse3.java

E:\Java Programs>java reverse3
Enter the no. :
100
Reverse of 100 is :
001
```

<u>RESULT:</u>

The JAVA program to reverse a number using for, while and recursion was implemented and executed successfully.

Exp No.: 2

Programs using classes and objects in Java

1. Write a java program to calculate average of n marks using array.

AIM:

To write a JAVA program to calculate average of n marks using array.

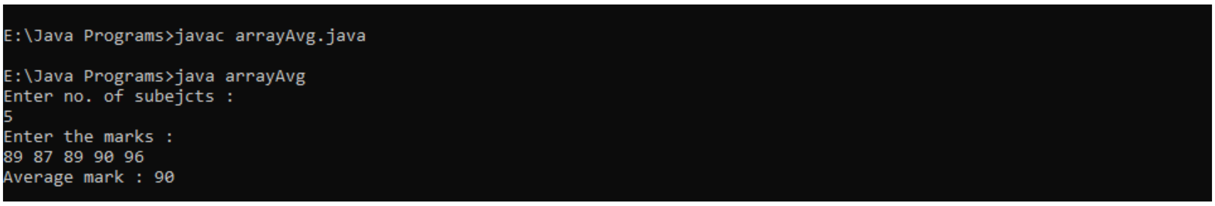
ALGORITHM:

```
Step 1: Start.
Step 2: Import required packages.
Step 3: Get the number of subjects and their corresponding marks from the user.
Step 4: Calculate the average of the marks.
Step 5: End.
```

PROGRAM:

```java
import java.util.Scanner;
public class arrayAvg
{
    public static void main(String[] args)
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter no. of subejcts : ");
        int n=s.nextInt();
        int arr[]=new int[n];
        int sum=0;
        System.out.println("Enter the marks : ");
        for(int i=0;i<n;i++)
        {
            arr[i]=s.nextInt();
            sum+=arr[i];
        }
        System.out.println("Average mark : "+sum/n);
        s.close();;
    }
}
```

OUTPUT:



```
E:\Java Programs>javac arrayAvg.java

E:\Java Programs>java arrayAvg
Enter no. of subejcts :
5
Enter the marks :
89 87 89 90 96
Average mark : 90
```

RESULT:

The JAVA program to calculate the average of n marks using array was implemented and executed successfully.

2. Write a java program to add and multiply two given matrices.

AIM:

To write a JAVA program to add and multiply to given matrices.

ALGORITHM:

Step 1: Start.

Step 2: Import required packages.

Step 3: Get the two matrices and the number of rows and columns in them from the user.

Step 4: Display the list of operations to the user and get their choice.

Step 5: If the choice is 1, the matrices are added and the resultant is printed.

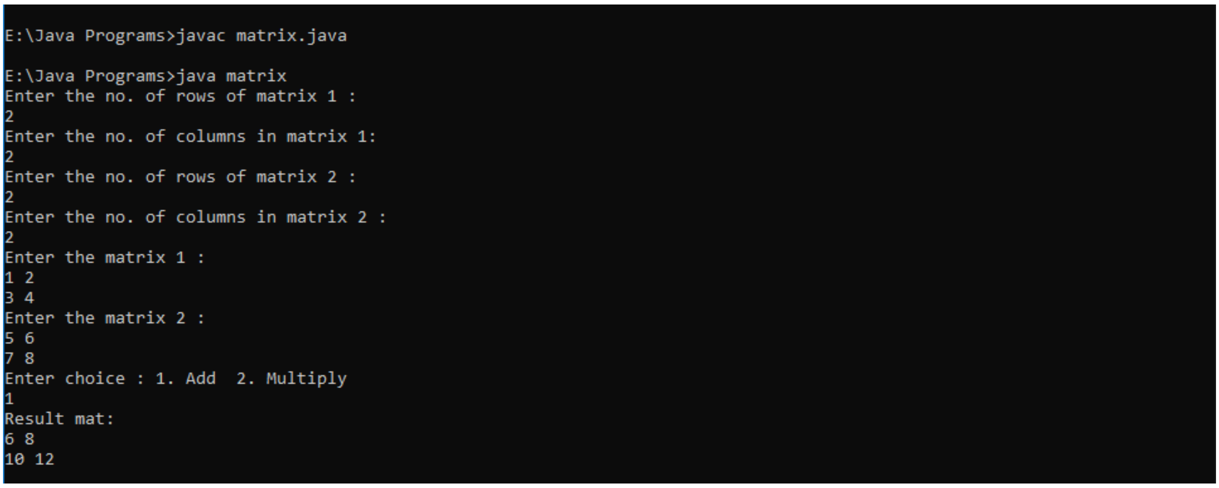
Step 6: If the choice is 2, the matrices are multiplied and the resultant is printed.

Step 7: End.

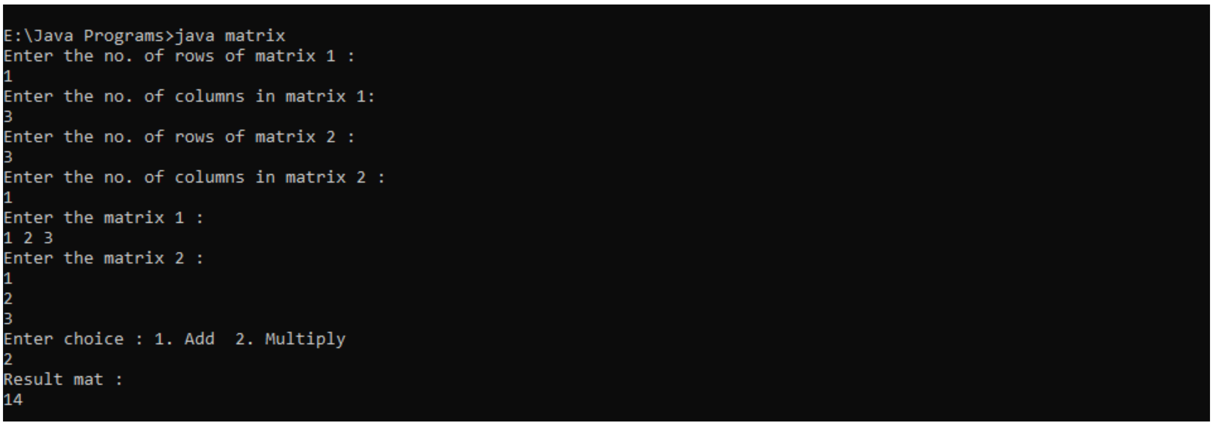
PROGRAM:

```java
import java.util.Scanner;
public class matrix
{
    static void getmat(int[][] mat,int r,int c)
    {
        Scanner s=new Scanner(System.in);
        int i,j;
        for(i=0;i<r;i++)
        {
            for(j=0;j<c;j++)
                mat[i][j]=s.nextInt();
        }
    }
    static void multiplication(int [][]mat1,int[][] mat2,int mat3[][],int r1,int c1,int r2,int c2)
    {
        int i,j,k;
        for(i=0;i<r1;i++)
        {
            for(j=0;j<c2;j++)
            {
                int sum=0;
                for(k=0;k<c1;k++)
                    sum+=(mat1[i][k]*mat2[k][j]);
                mat3[i][j]=sum;
            }
        }
    }
    static void addition(int[][] A,int[][] B,int[][] C,int r,int c)
    {
        for(int i=0;i<r;i++)
        {
            for(int j=0;j<c;j++)
                C[i][j]=A[i][j]+B[i][j];
        }
    }
    static void printmat(int[][] mat,int r,int c)
    {
        int i,j;
        for(i=0;i<r;i++)
        {
            for(j=0;j<c;j++)
                System.out.print(mat[i][j]+" ");
            System.out.println();
        }
    }
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        int r1,c1,r2,c2,result;
        System.out.println("Enter the no. of rows of matrix 1 :");
        r1=s.nextInt();
        System.out.println("Enter the no. of columns in matrix 1:");
        c1=s.nextInt();
        System.out.println("Enter the no. of rows of matrix 2 :");
        r2=s.nextInt();
        System.out.println("Enter the no. of columns in matrix 2 :");
        c2=s.nextInt();
        int A[][]=new int[r1][c1];
        int B[][]=new int[r2][c2];
        System.out.println("Enter the matrix 1 :");
        getmat(A,r1,c1);
        System.out.println("Enter the matrix 2 :");
        getmat(B,r2,c2);
        System.out.println("Enter choice : 1. Add 2. Multiply");
        int ch=s.nextInt();
        if(ch==1)
        {
            if(r1==r2 && c1==c2)
            {
                int C[][]=new int[r1][c1];
                addition(A,B,C,r1,c1);
                System.out.println("Result mat: ");
                printmat(C,r1,c1);
            }
            else
                System.out.println("Matrix mismatch");
        }
        else
        {
            if(c1==r2)
            {
                int[][] D=new int[r1][c2];
                multiplication(A,B,D,r1,c1,r2,c2);
                System.out.println("Result mat : ");
                printmat(D,r1,c2);
            }
            else
                System.out.println("Matrix mismatch");
        }

        s.close();
    }
}
```

OUTPUT:



```
E:\Java Programs>javac matrix.java

E:\Java Programs>java matrix
Enter the no. of rows of matrix 1 :
2
Enter the no. of columns in matrix 1:
2
Enter the no. of rows of matrix 2 :
2
Enter the no. of columns in matrix 2 :
2
Enter the matrix 1 :
1 2
3 4
Enter the matrix 2 :
5 6
7 8
Enter choice : 1. Add  2. Multiply
1
Result mat:
6 8
10 12
```



```
E:\Java Programs>java matrix
Enter the no. of rows of matrix 1 :
1
Enter the no. of columns in matrix 1:
3
Enter the no. of rows of matrix 2 :
3
Enter the no. of columns in matrix 2 :
1
Enter the matrix 1 :
1 2 3
Enter the matrix 2 :
1
2
3
Enter choice : 1. Add  2. Multiply
2
Result mat :
14
```

RESULT:

The JAVA program to add and multiply two given matrices was implemented successfully.

Exp No.: 3
Date:

Execute Java programs with various string handling methods

1. Write a java program that checks whether a given string is palindrome or not.

AIM:

To write a JAVA program that checks whether a given string is palindrome or not.

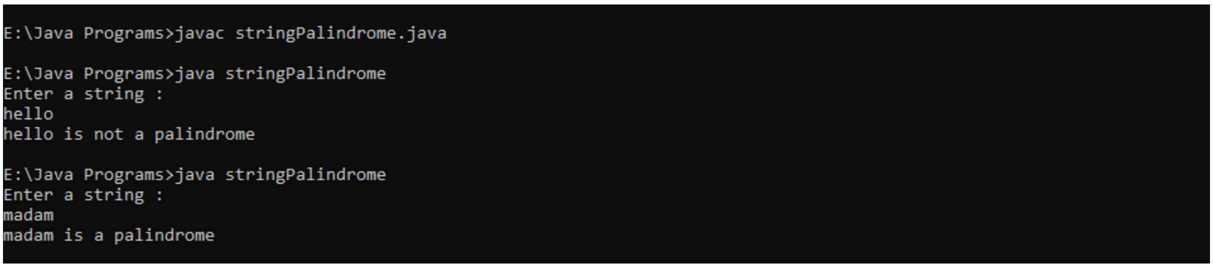
ALGORITHM:

Step 1: Start.
Step 2: Import required packages.
Step 3: Get the string from the user.
Step 4: If the string is equal to its reverse, the string is printed as a palindrome.
Step 5: Else, the string is printed as not a palindrome.
Step 6: End.

PROGRAM:

```java
import java.util.Scanner;
public class stringPalindrome
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter a string :");
        String str=s.nextLine();
        int l=str.length();
        String rev="";
        for(int i=l-1;i>=0;i--)
            rev+=str.charAt(i);
        if(str.equals(rev))
            System.out.println(str+" is a palindrome");
        else
            System.out.println(str+" is not a palindrome");
        s.close();
    }
}
```

OUTPUT:



```
E:\Java Programs>javac stringPalindrome.java

E:\Java Programs>java stringPalindrome
Enter a string :
hello
hello is not a palindrome

E:\Java Programs>java stringPalindrome
Enter a string :
madam
madam is a palindrome
```

RESULT:

The JAVA program that checks whether a given string is a palindrome or not was implemented and executed successfully.

2. Write a program to count vowels and consonants in a given string.

AIM:

To write a JAVA program to count vowels and consonants in a given string.

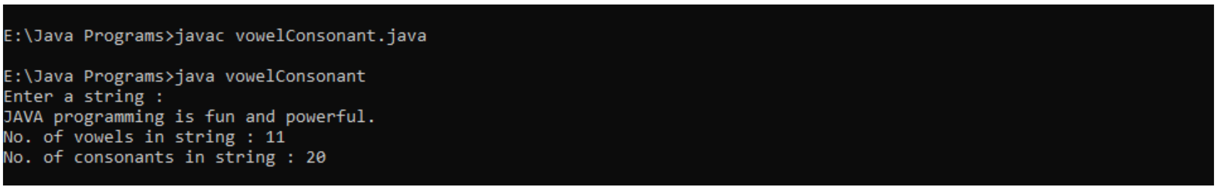
ALGORITHM:

Step 1: Start.
Step 2: Import required packages.
Step 3: Get the string from the user.
Step 4: Calculate the number of vowels and consonants in the string.
Step 5: Print the determined values.
Step 6: End.

PROGRAM:

```java
import java.util.Scanner;
public class vowelConsonant
{
    static boolean isVowel(char c)
    {
        char t=Character.toLowerCase(c);
        if(t=='a'||t=='e'||t=='i'||t=='o'||t=='u')
            return true;
        else
            return false;
    }
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter a string :");
        String str=s.nextLine();
        int vc=0,cc=0;
        for(int i=0;i<str.length();i++)
        {
            if(Character.isLetter(str.charAt(i)))
            {
                if(isVowel(str.charAt(i)))
                    vc++;
                else
                    cc++;
            }
        }
        System.out.println("No. of vowels in string : "+vc);
        System.out.println("No. of consonants in string : "+cc);
        s.close();
    }
}
```

OUTPUT:



```
E:\Java Programs>javac vowelConsonant.java

E:\Java Programs>java vowelConsonant
Enter a string :
JAVA programming is fun and powerful.
No. of vowels in string : 11
No. of consonants in string : 20
```

RESULT:

The JAVA program to count the number of vowels and consonants in a string was implemented and executed successfully.

3. Write a java program to sort strings in alphabetical order.

AIM:

To write a JAVA program to sort string in alphabetical order.

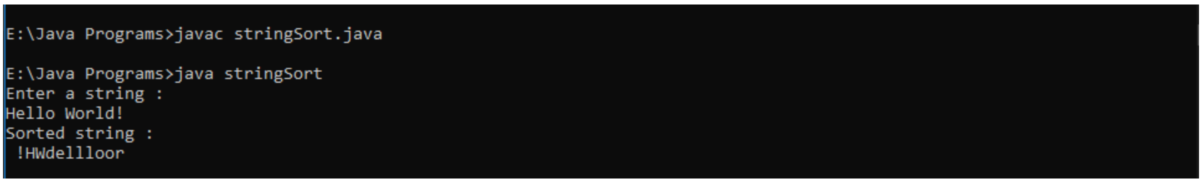
ALGORITHM:

```
Step 1: Start.
Step 2: Import required packages.
Step 3: Get the required string from the user.
Step 4: Sort the characters in the string in alphabetical order.
Step 5: Print the sorted string.
Step 6: End;
```

PROGRAM:

```java
import java.util.Scanner;
public class stringSort
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter a string :");
        String str=s.nextLine();
        char[] ch=str.toCharArray();
        int l=str.length();
        for(int i=0;i<l-1;i++)
        {
            for(int j=i+1;j<l;j++)
            {
                if(ch[i]>ch[j])
                {
                    char t=ch[i];
                    ch[i]=ch[j];
                    ch[j]=t;
                }
            }
        }
        System.out.println("Sorted string :");
        for(int i=0;i<l;i++)
            System.out.print(ch[i]);
        System.out.println();
        s.close();
    }
}
```

OUTPUT:



```
E:\Java Programs>javac stringSort.java

E:\Java Programs>java stringSort
Enter a string :
Hello World!
Sorted string :
 !HWdellloor
```

RESULT:

The JAVA program to sort string in alphabetical order was implemented and executed successfully.

4. Write a java program to find the occurrence of a character in a string.

AIM:

To write a JAVA program to find the occurrence of a character in a string.

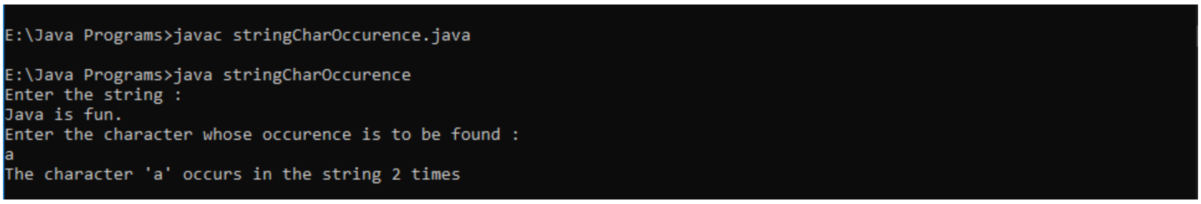
ALGORITHM:

Step 1: Start.
Step 2: Import required packages.
Step 3: Get the string and the character whose occurrence is to be found.
Step 4: Iterate through the string and count the occurrence of the given character.
Step 5: Print the number of times the character is present in the string.
Step 6: End.

PROGRAM:

```java
import java.util.Scanner;
public class stringCharOccurence
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the string :");
        String str=s.nextLine();
        System.out.println("Enter the character whose occurence is to be found
 :");
        String c=s.nextLine();
        if(str.contains(c))
        {
            int count=0;
            for(int i=str.indexOf(c);i<str.length();i++)
            {
                if(str.charAt(i)==c.charAt(0))
                    count++;
            }
            System.out.println("The character '"+c+"' occurs in the string "+c
ount+" times");
        }
        else
            System.out.println("The string doesn't contain character '"+c+"'")
;
        s.close();
    }
}
```

OUTPUT:



```
E:\Java Programs>javac stringCharOccurence.java

E:\Java Programs>java stringCharOccurence
Enter the string :
Java is fun.
Enter the character whose occurence is to be found :
a
The character 'a' occurs in the string 2 times
```

RESULT:

The JAVA program to find the occurrence of a character in string was implemented and executed successfully.

5. Write a java program to check whether a given string contains upper case values.

AIM:

To write a java program to check whether a given string contains upper case values.

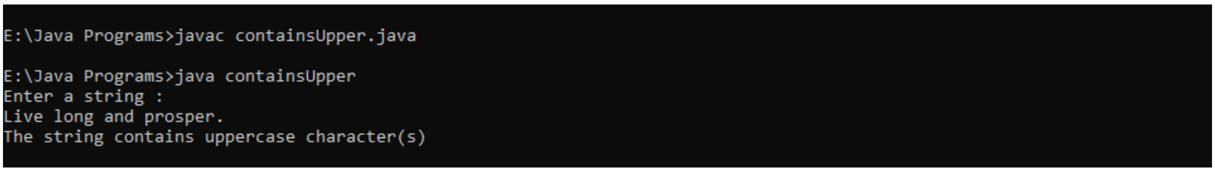
ALGORITHM:

Step 1: Start.
Step 2: Import required packages.
Step 3: Get the string from the user.
Step 4: Iterate through the string to check for upper case values.
Step 5: If an upper-case value is found, print that th string contains upper case values.
Step 6: Else, print the string doesn’t have any upper-case values.
Step 7: End.

PROGRAM:

```
import java.util.Scanner;
public class containsUpper
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter a string :");
        String str=s.nextLine();
        int i,flag=0;
        for(i=0;i<str.length();i++)
        {
            if(Character.isUpperCase(str.charAt(i)))
            {
                flag=1;
                break;
            }
        }
        if(flag==0)
            System.out.println("The string doesn't have any uppercase characte
rs");
        else
            System.out.println("The string contains uppercase character(s)");
        s.close();
    }

}
```

OUTPUT:



```
E:\Java Programs>javac containsUpper.java

E:\Java Programs>java containsUpper
Enter a string :
Live long and prosper.
The string contains uppercase character(s)
```

RESULT:

The JAVA program to check whether a string contains upper case values was implemented and executed successfully.

6. Write a java program to demonstrate:
a) Joining of two strings.
b) Comparing two strings.
c) Extracting substrings.
d) Splitting string into substrings.

a) Joining of two strings:

AIM:

To write a JAVA program to demonstrate joining of two strings.

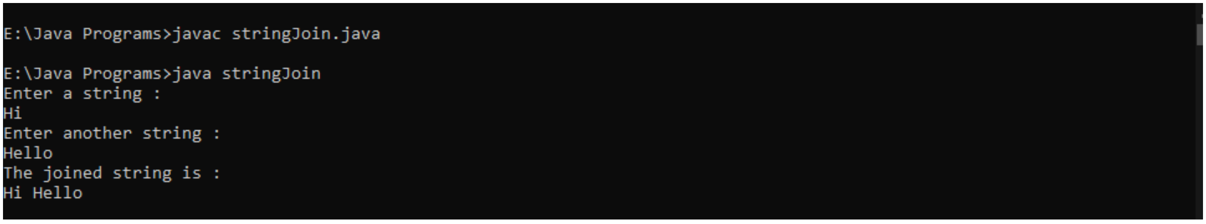
ALGORITHM:

Step 1: Start.
Step 2: Import required packages.
Step 3: Get the two strings to be joined from the user.
Step 4: Join the two strings using join().
Step 5: Print the joined string.
Step 6: End.

PROGRAM:

```java
import java.util.Scanner;
public class stringJoin
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter a string :");
        String str1=s.nextLine();
        System.out.println("Enter another string :");
        String str2=s.nextLine();
        String str3=String.join(" ",str1,str2);
        System.out.println("The joined string is :");
        System.out.println(str3);
        s.close();
    }
}
```

OUTPUT:



```
E:\Java Programs>javac stringJoin.java

E:\Java Programs>java stringJoin
Enter a string :
Hi
Enter another string :
Hello
The joined string is :
Hi Hello
```

RESULT:

The JAVA program to demonstrate joining two strings was implemented and executed successfully.

b) Comparing two strings

AIM:

To write a JAVA program to demonstrate comparing two strings.

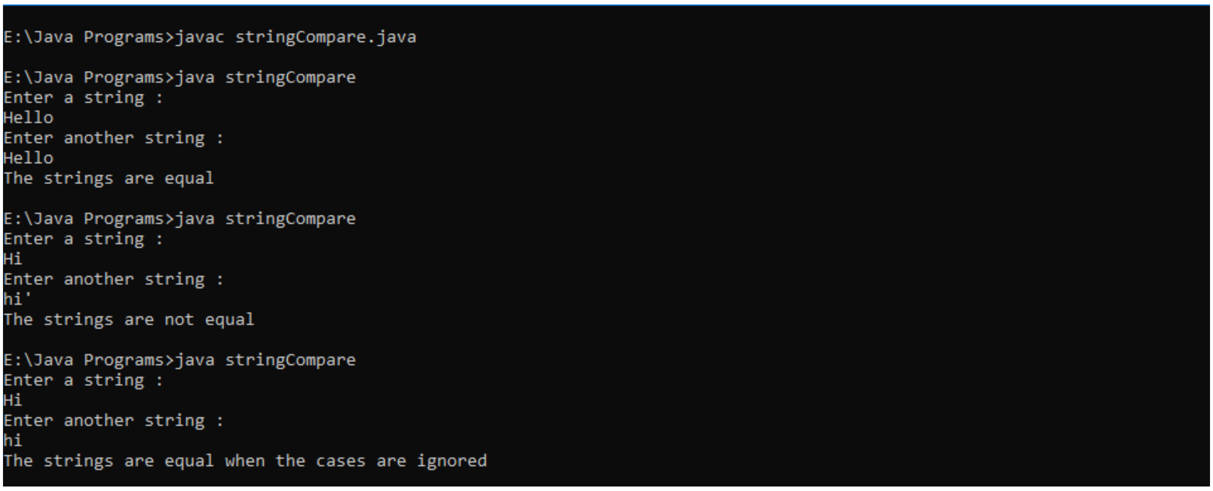
ALGORITHM:

Step 1: Start.
Step 2: Import required packages.
Step 3: Get the strings from the user.
Step 4: Check if the strings are equal and print out the findings.
Step 5: End.

PROGRAM:

```java
import java.util.Scanner;
public class stringCompare
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter a string :");
        String str1=s.nextLine();
        System.out.println("Enter another string :");
        String str2=s.nextLine();
        if(str1.equals(str2))
            System.out.println("The strings are equal");
        else if(str1.equalsIgnoreCase(str2))
            System.out.println("The strings are equal when the cases are ignor
ed");
        else
            System.out.println("The strings are not equal");
        s.close();
    }

}
```

OUTPUT:



```
E:\Java Programs>javac stringCompare.java

E:\Java Programs>java stringCompare
Enter a string :
Hello
Enter another string :
Hello
The strings are equal

E:\Java Programs>java stringCompare
Enter a string :
Hi
Enter another string :
hi'
The strings are not equal

E:\Java Programs>java stringCompare
Enter a string :
Hi
Enter another string :
hi
The strings are equal when the cases are ignored
```

RESULT:

The JAVA program to demonstrate comparing two strings was implemented and executed successfully.

c) Extracting substrings

AIM:

To write a JAVA program to demonstrate extracting substrings.

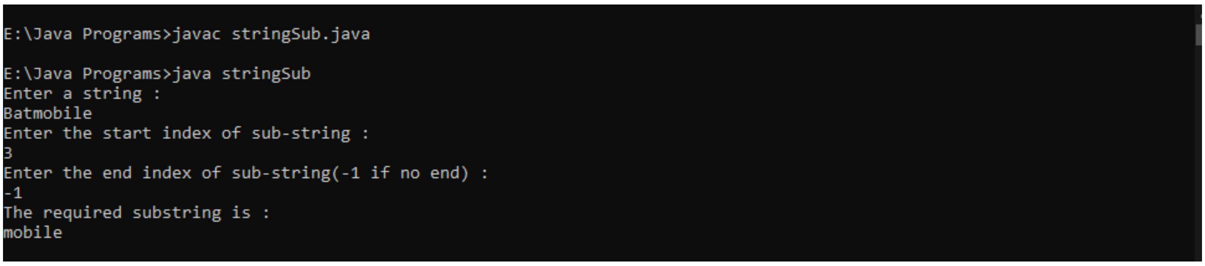
ALGORITHM:

Step 1: Start.
Step 2: Import required packages.
Step 3: Get the string from the user.
Step 4: Get the start and end index of the required substring.
Step 5: Print the desired substring.
Step 6: End.

PROGRAM:

```
import java.util.Scanner;
public class stringSub
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter a string :");
        String str=s.nextLine();
        System.out.println("Enter the start index of sub-string :");
        int b=s.nextInt();
        System.out.println("Enter the end index of sub-string(-
1 if no end) :");
        int e=s.nextInt();
        String sub;
        if(b>=str.length()||b<0||e>=str.length())
            System.out.println("Index out of range.");
        else if(e==-1)
        {
            sub=str.substring(b);
            System.out.println("The required substring is :");
            System.out.println(sub);
        }
        else
        {
            sub=str.substring(b,e);
            System.out.println("The required substring is :");
            System.out.println(sub);
        }
        s.close();
    }
}
```

OUTPUT:



```
E:\Java Programs>javac stringSub.java

E:\Java Programs>java stringSub
Enter a string :
Batmobile
Enter the start index of sub-string :
3
Enter the end index of sub-string(-1 if no end) :
-1
The required substring is :
mobile
```

RESULT:

The JAVA program to demonstrate extracting substring was implemented and executed successfully.

d) Splitting the string into substrings

AIM:

To write a JAVA program to demonstrate splitting the string into substrings.

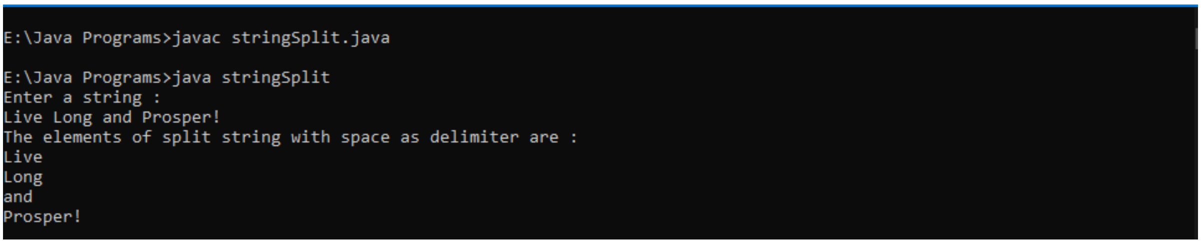
ALGORITHM:

```
Step 1: Start.
Step 2: Import required packages.
Step 3: Get the string from the user.
Step 4: Split the string at spaces.
Step 5: Print the split substrings.
Step 6: End.
```

PROGRAM:

```java
import java.util.Scanner;
public class stringSplit
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter a string :");
        String str=s.nextLine();
        String[] strSplit=str.split(" ");
        System.out.println("The elements of split string with space as delimit
er are :");
        for(int i=0;i<strSplit.length;i++)
            System.out.println(strSplit[i]);
        s.close();
    }
}
```

OUTPUT:



```
E:\Java Programs>javac stringSplit.java

E:\Java Programs>java stringSplit
Enter a string :
Live Long and Prosper!
The elements of split string with space as delimiter are :
Live
Long
and
Prosper!
```

RESULT:

The JAVA program to demonstrate splitting the string into substrings was implemented and executed successfully.

Exp No.: 4a

Date:

Programs Using Classes and Constructors in JAVA

1. Write a program to print the area and perimeter of a triangle having sides of 3, 4 and 5 units by creating a class named 'Triangle' without any parameter in its constructor.

AIM:

To write a JAVA program to print the area & perimeter of a triangle having sides of 3, 4, 5.

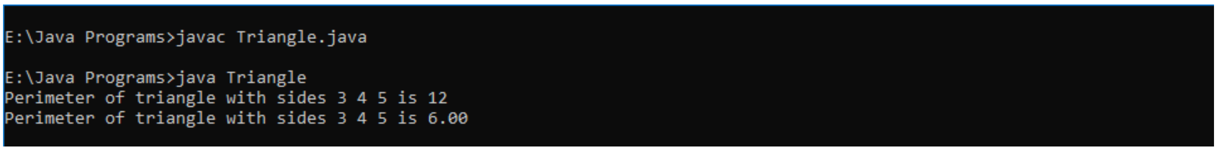
ALGORITHM:

```
    Step 1: Start.
    Step 2: Import required packages.
    Step 3: Object for the triangle is created and the sides are assigned as 3, 4, and 5 respectively.
    Step 4: Print the area and perimeter of the triangle
    Step 5: End.
```

PROGRAM:

```java
public class Triangle
{
    private int s1,s2,s3;
    Triangle()
    {
        s1=3;
        s2=4;
        s3=5;
    }
    public int perimeter()
    {
        return s1+s2+s3;
    }
    public double area()
    {
        double s=(s1+s2+s3)/2.0;
        return Math.sqrt(s*(s-s1)*(s-s2)*(s-s3));
    }
    public void sides()
    {
        System.out.print(s1+" "+s2+" "+s3);
    }
    public static void main(String args[])
    {
        Triangle t=new Triangle();
        System.out.print("Perimeter of triangle with sides ");
        t.sides();
        System.out.println(" is "+t.perimeter());
        System.out.print("Perimeter of triangle with sides ");
        t.sides();
        System.out.printf(" is %.2f\n",t.area());
    }
}
```

OUTPUT:



```
E:\Java Programs>javac Triangle.java

E:\Java Programs>java Triangle
Perimeter of triangle with sides 3 4 5 is 12
Perimeter of triangle with sides 3 4 5 is 6.00
```

RESULT:

The required JAVA program was implemented and executed successfully.

2. Print the sum, difference and product of two complex numbers by creating a class named 'Complex' with separate methods for each operation whose real and imaginary parts are entered by user.

AIM:

To write a JAVA program to print the sum, difference and product of two complex numbers.

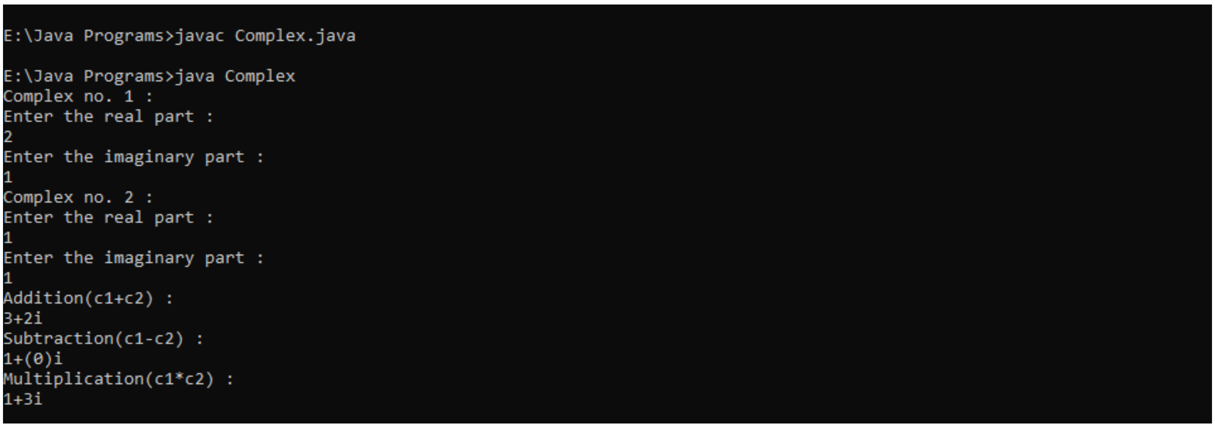
ALGORITHM:

Step 1: Start.
Step 2: Import required packages.
Step 3: Get the two complex numbers from the user.
Step 4: Perform the addition, subtraction and multiplication of the complex numbers and print result.
Step 5: End.

PROGRAM:

```
import java.util.Scanner;
public class Complex
{
    private int real,image;
    Complex(int real,int image)
    {
        this.real=real;
        this.image=image;
    }
    public void display()
    {
        System.out.print(real+"+"+image+"i");
    }
    public void add(Complex c)
    {
        System.out.println((real+c.real)+"+"+(image+c.image)+"i");
    }
    public void sub(Complex c)
    {
        System.out.println((real-c.real)+"+("+(image-c.image)+")i");
    }
    public void mult(Complex c)
    {
        int t1=real*c.real;
        int t2=image*c.image;
        System.out.println((t1-t2)+"+"+(t1+t2)+"i");
    }
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Complex no. 1 :");
        System.out.println("Enter the real part :");
        int real1=s.nextInt();
        System.out.println("Enter the imaginary part :");
        int image1=s.nextInt();
        Complex c1=new Complex(real1,image1);
        System.out.println("Complex no. 2 :");
        System.out.println("Enter the real part :");
        int real2=s.nextInt();
        System.out.println("Enter the imaginary part :");
        int image2=s.nextInt();
        Complex c2=new Complex(real2,image2);
        System.out.println("Addition(c1+c2) :");
        c1.add(c2);
        System.out.println("Subtraction(c1-c2) :");
        c1.sub(c2);
        System.out.println("Multiplication(c1*c2) :");
        c1.mult(c2);
        s.close();
    }
}
```

OUTPUT:



```
E:\Java Programs>javac Complex.java

E:\Java Programs>java Complex
Complex no. 1 :
Enter the real part :
2
Enter the imaginary part :
1
Complex no. 2 :
Enter the real part :
1
Enter the imaginary part :
1
Addition(c1+c2) :
3+2i
Subtraction(c1-c2) :
1+(0)i
Multiplication(c1*c2) :
1+3i
```

RESULT:

The required JAVA program was implemented and executed successfully.

3. Write a java program which creates class Student (Rollno, Name,- Number of subjects, Marks of each subject)(Number of subjects varies for each student) Write a parameterized constructor which initializes roll no, name & Number of subjects and create the array of marks dynamically. Display the details of all students with percentage and class obtained.

AIM:

To write a JAVA program with a class Student.

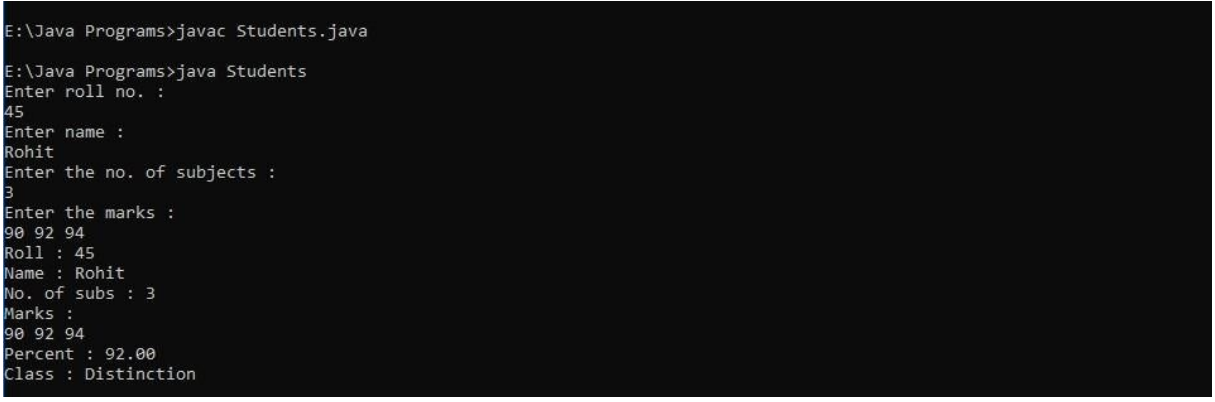
ALGORITHM:

Step 1: Start.
Step 2: Import required packages.
Step 3: Get the student details from the user.
Step 4: Evaluate the percentage and the class.
Step 5: Print the complete mark statement of the student.
Step 6: End.

PROGRAM:

```
import java.util.Scanner;
public class Students
{
    private int roll,nos,tot;
    private String name;
    private int[] marks;
    Students(int roll,String name,int nos,int[] marks)
    {
        this.roll=roll;
        this.name=name;
        this.nos=nos;
        this.marks=new int[nos];
        tot=0;
        for(int i=0;i<nos;i++)
        {
            this.marks[i]=marks[i];
            tot+=marks[i];
        }
    }
    public void display()
    {
        System.out.println("Roll : "+roll);
        System.out.println("Name : "+name);
        System.out.println("No. of subs : "+nos);
        System.out.println("Marks :");
        for(int i=0;i<nos;i++)
            System.out.print(marks[i]+" ");
        System.out.println();
        System.out.printf("Percent : %.2f\n",(tot*1.0)/nos);
        System.out.println("Class : "+getclass());
    }
    private String getclass()
    {
        double a=(tot*1.0)/nos;
        if(a>90)
            return "Distinction";
        else if(a>70)
            return "I Class";
        else if(a>=50)
            return "II Class";
        else
            return "Fail";

    }
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter roll no. :");
        int roll=s.nextInt();
        s.nextLine();
        System.out.println("Enter name :");
        String name=s.nextLine();
        System.out.println("Enter the no. of subjects : ");
        int nos=s.nextInt();
        int[] marks=new int[nos];
        System.out.println("Enter the marks :");
        for(int i=0;i<nos;i++)
            marks[i]=s.nextInt();
        Students st=new Students(roll,name,nos,marks);
        st.display();
        s.close();
    }
}
```

OUTPUT:



```
E:\Java Programs>javac Students.java

E:\Java Programs>java Students
Enter roll no. :
45
Enter name :
Rohit
Enter the no. of subjects :
3
Enter the marks :
90 92 94
Roll : 45
Name : Rohit
No. of subs : 3
Marks :
90 92 94
Percent : 92.00
Class : Distinction
```

RESULT:

The required JAVA program was implemented and executed successfully.

Exp No.: 4b
Date:

Design programs using all the types of inheritances

1. The classes Programmer and the Database Pro both extend the Employee class and they inherit the fields name, age and salary from employee. Both programmers and database professionals can inherit all the attributes they need from the class Employee, but they need to keep their own special attributes in their own classes.

Design for three classes and implement it.

Read data and print data for some employees.

Aim:

To write a JAVA program with classes Programmer and Database Pro, which extends from class Employee.

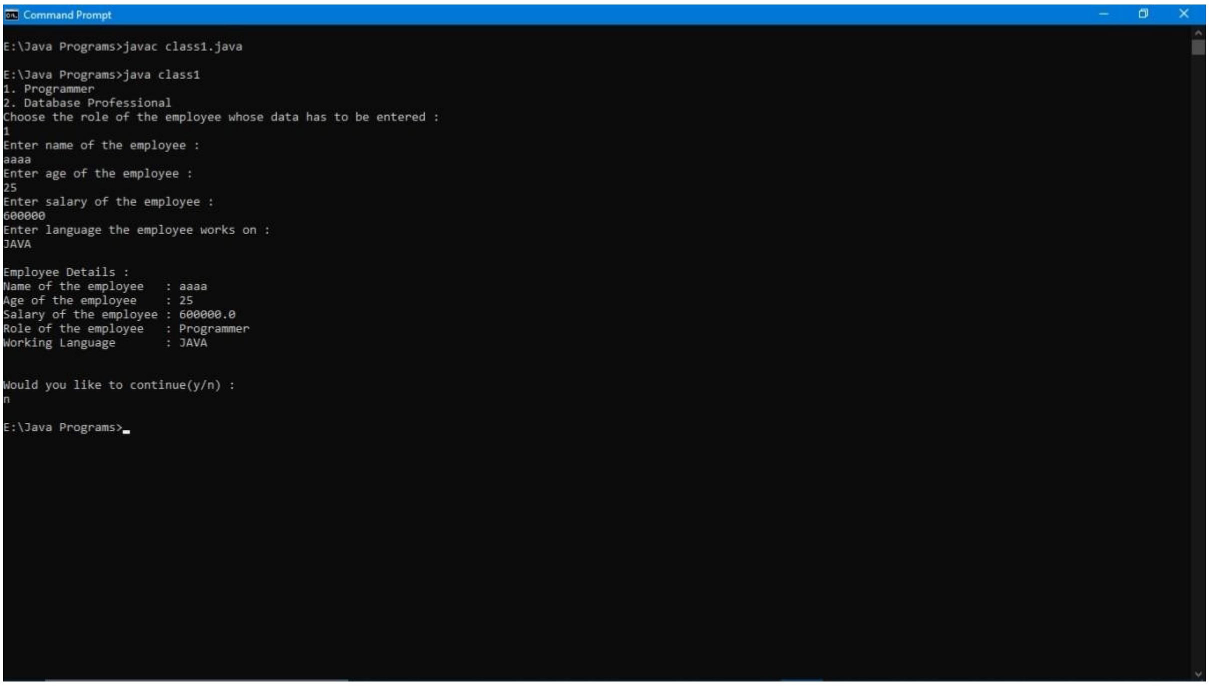
Algorithm:

```
    Step 1: Start.
    Step 2: Import required packages.
    Step 3: Display the two posts and get the user's choice.
    Step 4: If the choice is 1, get the details for Programmer, and display the complete details.
    Step 5: If the choice is 2, get the details for Database Professional, and display the complete details.
    Step 6: Repeat step 3-5, if the user wants to continue.
    Step 7: End.
```

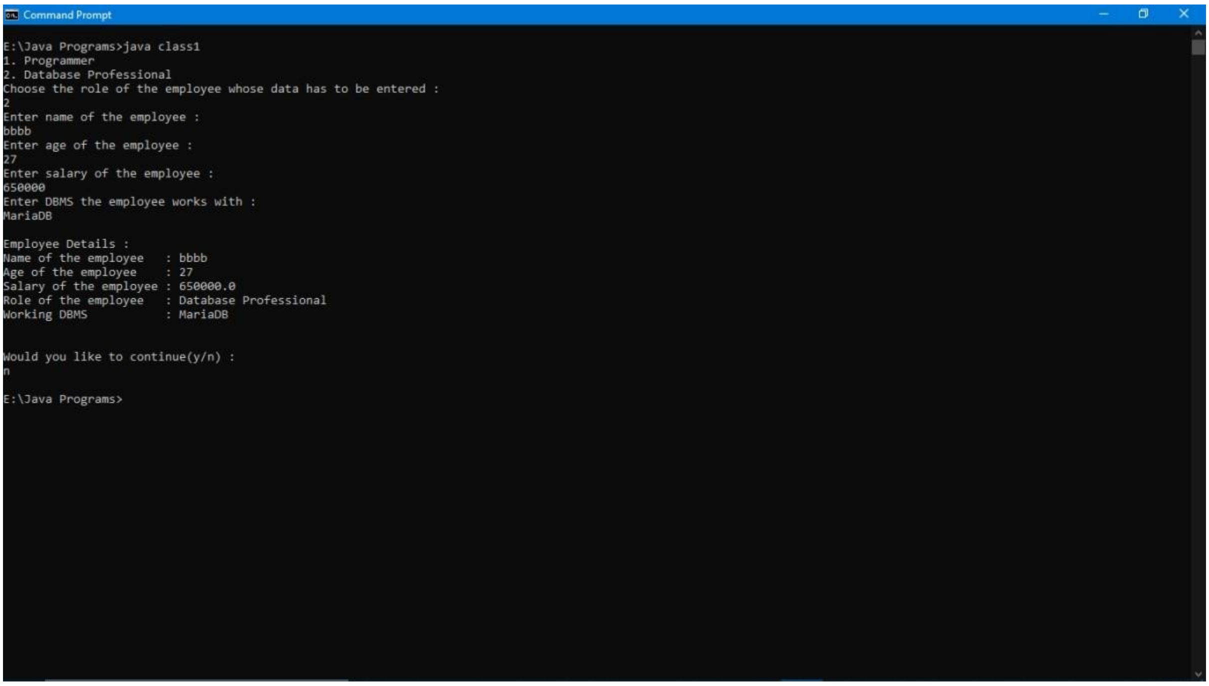
Program:

```
import java.util.Scanner;
class Employee
{
    protected String name;
    protected int age;
    protected double salary;
    Employee(String name,int age,double salary)
    {
        this.name=name;
        this.age=age;
        this.salary=salary;
    }
    public void printEmp()
    {
        System.out.println("\nEmployee Details : ");
        System.out.println("Name of the employee    : "+this.name);
        System.out.println("Age of the employee     : "+this.age);
        System.out.println("Salary of the employee : "+this.salary);
    }
}
class Programmer extends Employee
{
    final static String role="Programmer";
    private String lang;
    Programmer(String name,int age,double salary,String lang)
    {
        super(name,age,salary);
        this.lang=lang;
    }
    public void printProg()
    {
        super.printEmp();
        System.out.println("Role of the employee    : "+Programmer.role);
        System.out.println("Working Language        : "+this.lang);
    }
}


class DatabasePro extends Employee
{
    final static String role="Database Professional";
    private String dbms;
    DatabasePro(String name,int age,double salary,String dbms)
    {
        super(name,age,salary);
        this.dbms=dbms;
    }
    public void printDataPro()
    {
        super.printEmp();
        System.out.println("Role of the employee    : "+DatabasePro.role);
        System.out.println("Working DBMS            : "+this.dbms);
    }
}
public class class1
{
    public static void main(String[] agrs)
    {
        Scanner s=new Scanner(System.in);
        char c;
        do
        {
            System.out.println("1. Programmer");
            System.out.println("2. Database Professional");
            System.out.println("Choose the role of the employee whose data has
 to be entered : ");
            int ch=s.nextInt();
            s.nextLine();
            if(ch==1||ch==2)
            {
                System.out.println("Enter name of the employee : ");
                String name=s.nextLine();
                System.out.println("Enter age of the employee : ");
                int age=s.nextInt();
                System.out.println("Enter salary of the employee : ");
                double salary=s.nextDouble();
                s.nextLine();
                if(ch==1)
                {
                    System.out.println("Enter language the employee works on :
 ");

                    String lang=s.nextLine();
                    Programmer p=new Programmer(name,age,salary,lang);
                    p.printProg();
                }

                else
                {
                    System.out.println("Enter DBMS the employee works with : "
);

                    String dbms=s.nextLine();
                    DatabasePro d=new DatabasePro(name,age,salary,dbms);
                    d.printDataPro();
                }
            }
            else

                System.out.println("Invalid Command!");
            System.out.println("\n\nWould you like to continue(y/n) : ");
            c=s.nextLine().charAt(0);
        }while(c=='y'||c=='Y');
        s.close();
    }
}
```

Output:



```
E:\Java Programs>javac class1.java

E:\Java Programs>java class1
1. Programmer
2. Database Professional
Choose the role of the employee whose data has to be entered :
1
Enter name of the employee :
aaaa
Enter age of the employee :
25
Enter salary of the employee :
600000
Enter language the employee works on :
JAVA

Employee Details :
Name of the employee   : aaaa
Age of the employee    : 25
Salary of the employee : 600000.0
Role of the employee   : Programmer
Working Language       : JAVA


Would you like to continue(y/n) :
n

E:\Java Programs>
```

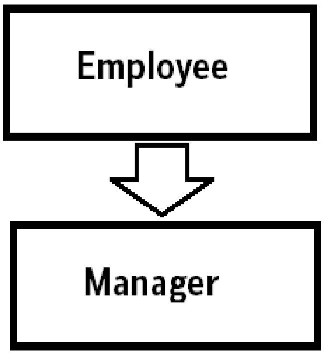


```
E:\Java Programs>java class1
1. Programmer
2. Database Professional
Choose the role of the employee whose data has to be entered :
2
Enter name of the employee :
bbbb
Enter age of the employee :
27
Enter salary of the employee :
650000
Enter DBMS the employee works with :
MariaDB

Employee Details :
Name of the employee   : bbbb
Age of the employee    : 27
Salary of the employee : 650000.0
Role of the employee   : Database Professional
Working DBMS           : MariaDB


Would you like to continue(y/n) :
n

E:\Java Programs>
```

Result:

The required JAVA program was written and executed successfully.

2. Design classes, implement and test it



Aim:

To write a JAVA program with class Manager which inherits class Employee.

Algorithm:

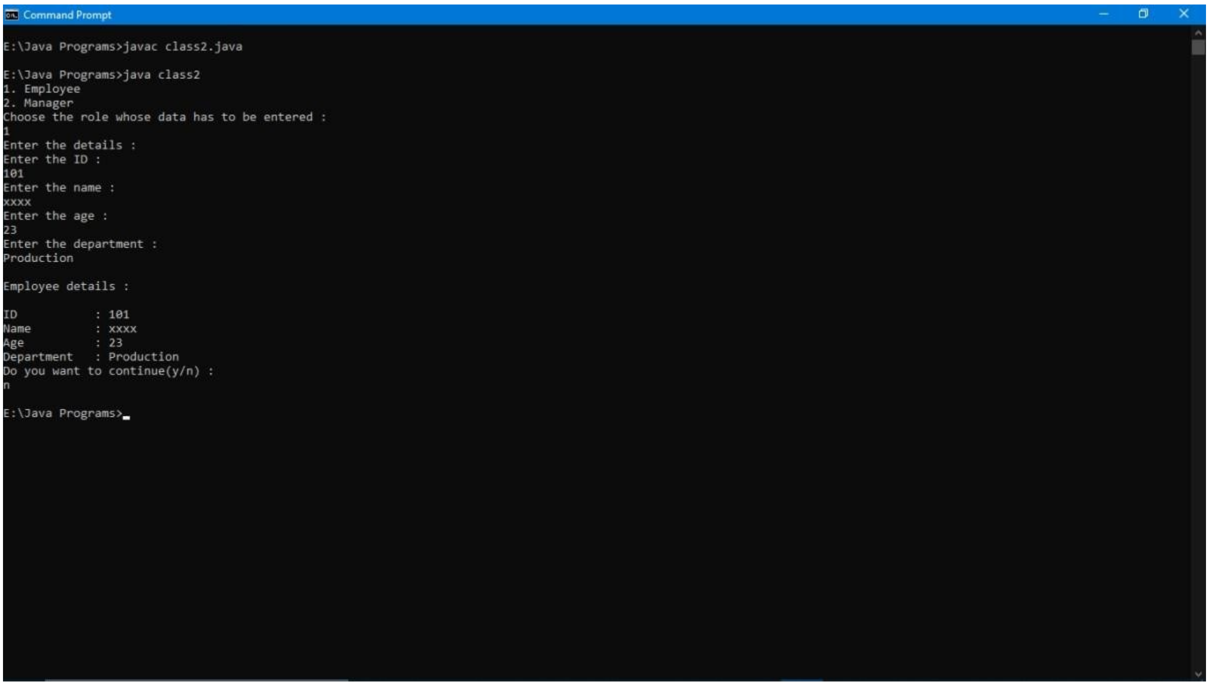
Step 1: Start.
Step 2: Import required packages.
Step 3: Display the two roles and get the user’s choice.
Step 4: If the choice is 1, get the details of the Employee and display the complete details.
Step 5: If the choice is 2, get the details of the Manager and display the complete details.
Step 6: Repeat steps 3-5, if the user wants to continue.
Step 7: End.

Program:

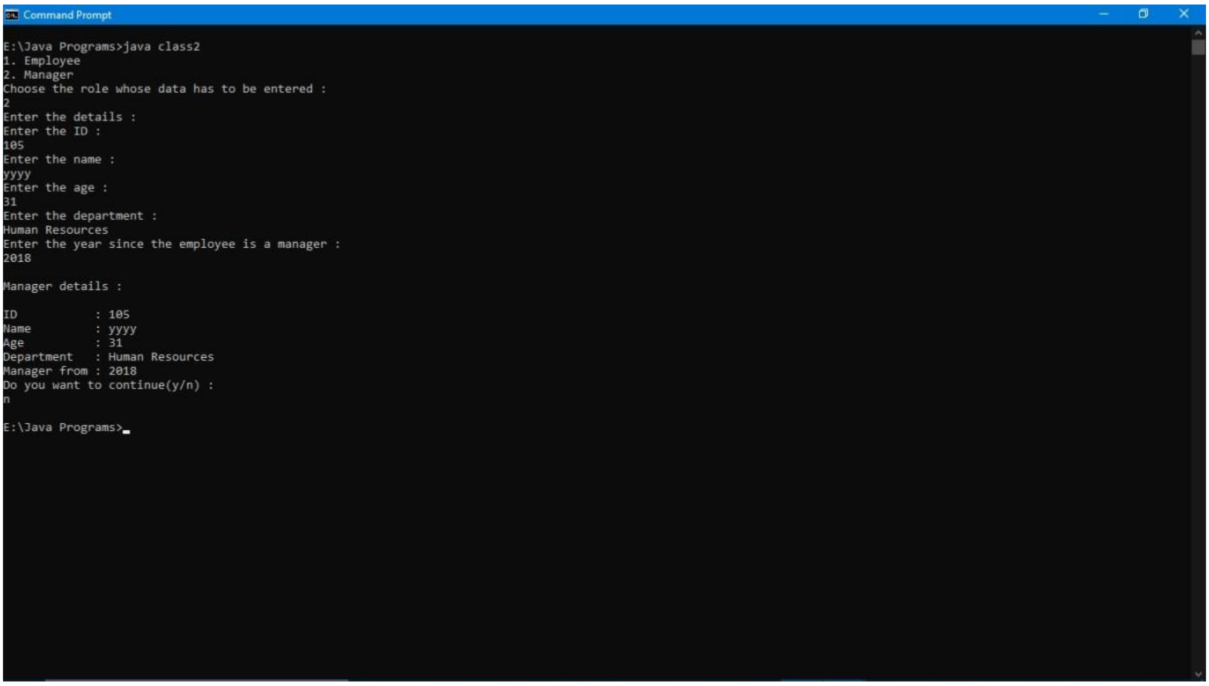
```
import java.util.Scanner;
class Employee
{
    protected int id,age;
    protected String name,dept;
    Employee(int id,String name,int age,String dept)
    {
        this.id=id;
        this.name=name;
        this.age=age;
        this.dept=dept;
    }
    public void dispEmp()
    {
        System.out.println("ID           : "+this.id);
        System.out.println("Name         : "+this.name);
        System.out.println("Age          : "+this.age);
        System.out.println("Department   : "+this.dept);
    }
}
```

```java
class Manager extends Employee
{
    protected int mgr_from;
    Manager(int id,String name,int age,String dept,int mgr_from)
    {
        super(id,name,age,dept);
        this.mgr_from=mgr_from;
    }
    public void dispMgr()
    {
        super.dispEmp();
        System.out.println("Manager from : "+this.mgr_from);
    }
}
public class class2
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        char c;
        do
        {
            System.out.println("1. Employee");
            System.out.println("2. Manager");
            System.out.println("Choose the role whose data has to be entered :
 ");

            int ch=s.nextInt();
            System.out.println("Enter the details :");
            if(ch==1||ch==2)
            {
                System.out.println("Enter the ID : ");
                int id=s.nextInt();
                s.nextLine();
                System.out.println("Enter the name : ");
                String name=s.nextLine();
                System.out.println("Enter the age : ");
                int age=s.nextInt();
                s.nextLine();
                System.out.println("Enter the department : ");
                String dept=s.nextLine();
                if(ch==1)
                {
                    Employee e=new Employee(id,name,age,dept);
                    System.out.println("\nEmployee details : \n");
                    e.dispEmp();
                }
                else
                {
                    System.out.println("Enter the year since the employee is a
 manager : ");
                    int mgr_from=s.nextInt();
                    s.nextLine();
                    Manager m=new Manager(id,name,age,dept,mgr_from);
                    System.out.println("\nManager details : \n");
                    m.dispMgr();
                }
            }
            else

                System.out.println("Invalid command!");
            System.out.println("Do you want to continue(y/n) : ");
            c=s.nextLine().charAt(0);
        }while(c=='y'||c=='Y');
        s.close();
    }
}
```

Output:



```
E:\Java Programs>javac class2.java

E:\Java Programs>java class2
1. Employee
2. Manager
Choose the role whose data has to be entered :
1
Enter the details :
Enter the ID :
101
Enter the name :
xxxx
Enter the age :
23
Enter the department :
Production

Employee details :

ID           : 101
Name         : xxxx
Age          : 23
Department   : Production
Do you want to continue(y/n) :
n

E:\Java Programs>
```



```
E:\Java Programs>java class2
1. Employee
2. Manager
Choose the role whose data has to be entered :
2
Enter the details :
Enter the ID :
105
Enter the name :
yyyy
Enter the age :
31
Enter the department :
Human Resources
Enter the year since the employee is a manager :
2018

Manager details :

ID           : 105
Name         : yyyy
Age          : 31
Department   : Human Resources
Manager from : 2018
Do you want to continue(y/n) :
n

E:\Java Programs>
```

Result:

The required JAVA program was written and executed successfully.

3. Write JAVA program to illustrate Method Overriding:
The method **draw()** is being overridden in the three subclasses **(Square, Circle, and Hexagon)** of their base class or superclass **Shape.**

Aim:

To write a JAVA program with classes Square, Circle and Hexagon which inherits the class Shape and overrides the method draw().

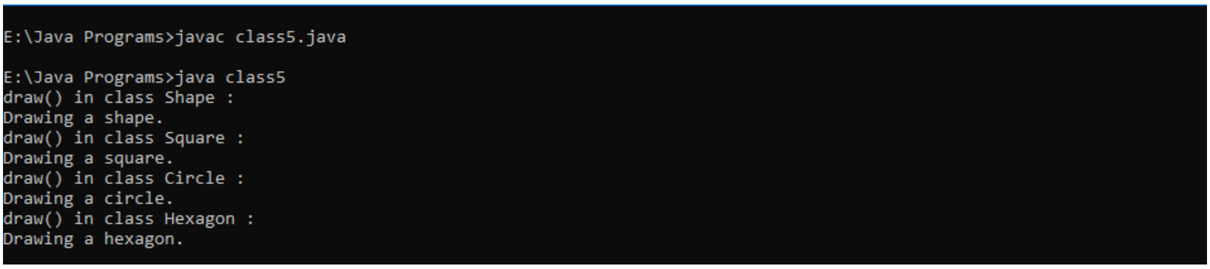
Algorithm:

```
Step 1: Start.
Step 2: Import required packages.
Step 3: Create an object for Square class and call its draw().
Step 4: Create an object for Circle class and call its draw().
Step 5: Create an object for Hexagon class and call its draw().
Step 6: End.
```

Program:

```java
class Shape
{
    public void draw()
    {
        System.out.println("Drawing a shape.");
    }
}
class Square extends Shape
{
    public void draw()
    {
        System.out.println("Drawing a square.");
    }
}
class Circle extends Shape
{
    public void draw()
    {
        System.out.println("Drawing a circle.");
    }
}
class Hexagon extends Shape
{
    public void draw()
    {
        System.out.println("Drawing a hexagon.");
    }
}
public class class5
{
    public static void main(String args[])
    {
        Shape s=new Shape();
        System.out.println("draw() in class Shape :");
        s.draw();
        s=new Square();
        System.out.println("draw() in class Square :");
        s.draw();
        s=new Circle();
        System.out.println("draw() in class Circle :");
        s.draw();
        s=new Hexagon();
        System.out.println("draw() in class Hexagon :");
        s.draw();
    }
}
```

Output:



```
E:\Java Programs>javac class5.java

E:\Java Programs>java class5
draw() in class Shape :
Drawing a shape.
draw() in class Square :
Drawing a square.
draw() in class Circle :
Drawing a circle.
draw() in class Hexagon :
Drawing a hexagon.
```

Result:

The required JAVA program was written and executed successfully.

Exp No.: 4c

Date:

Design programs using Interfaces

1. Develop an Interest interface which contains simpleInterest and compInterest methods and static final field of Rate 25%. Write a class to implement those methods.

AIM:

To write a JAVA program with an interface Interest.

ALGORITHM:

Step 1: Start.

Step 2: Display the list of choices and get the choice from the user.

Step 3: If the choice is 1, compute get the principle amount, time and rate interest and compute the simple interest.

Step 4: If the choice is 2, get the principle amount, time, rate of interest and no. of times the interest is applied and compute the compound interest.

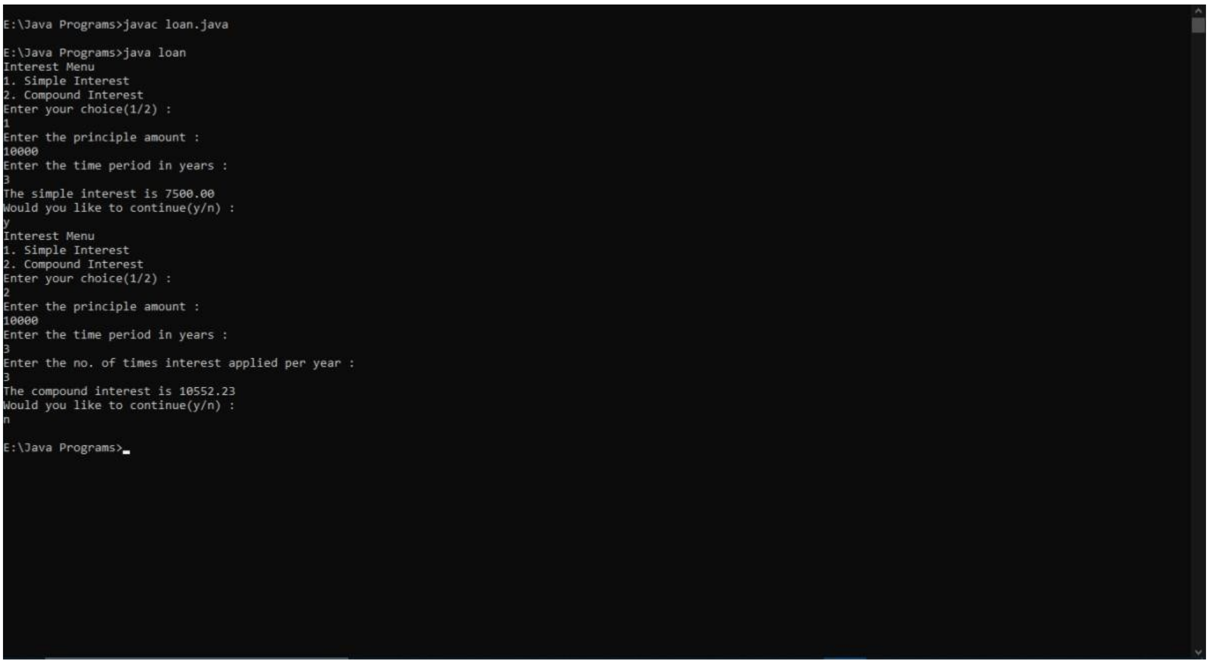
Step 5: Repeat the steps 2-4, if the user wants to continue.

Step 5: End.

PROGRAM:

```
import java.util.Scanner;
interface interest
{
    public double simpleinterest();
    public double compoundinterest();
    double rate=25;
}
public class loan implements interest
{
    private double p;
    private int t,n;
    loan(double p,int t)
    {
        this.p=p;
        this.t=t;
    }
    loan(double p,int t,int n)
    {
        this.p=p;
        this.t=t;
        this.n=n;
    }
    public double simpleinterest()
    {
        return p*rate*t/100;
    }
    public double compoundinterest()
    {
        double r=rate/100.0;
        return p*(Math.pow((1+(r/n)),(n*t)))-p;
    }
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        char  c='y';
        while(c=='y'||c=='Y')
        {
            System.out.println("Interest Menu");
            System.out.println("1. Simple Interest");
            System.out.println("2. Compound Interest");
            System.out.println("Enter your choice(1/2) : ");
            int ch=s.nextInt();
            if(ch==1||ch==2)
            {
                System.out.println("Enter the principle amount : ");
                double p=s.nextDouble();
                System.out.println("Enter the time period in years : ");
                int t=s.nextInt();
                s.nextLine();
                if(ch==1)
                {
                    loan si=new loan(p,t);
                    System.out.printf("The simple interest is %.2f\n",si.simpleinterest());
                }
                else
                {
                    System.out.println("Enter the no. of times interest applied per year : ");
                    int n=s.nextInt();
                    s.nextLine();
                    loan ci=new loan(p,t,n);
                    System.out.printf("The compound interest is %.2f\n",ci.compoundinterest());
                }
            }
            else
                System.out.println("Invalid command!");
            System.out.println("Would you like to continue(y/n) : ");
            c=s.nextLine().charAt(0);
        }
        s.close();
    }
}
```

OUTPUT:



```
E:\Java Programs>javac loan.java

E:\Java Programs>java loan
Interest Menu
1. Simple Interest
2. Compound Interest
Enter your choice(1/2) :
1
Enter the principle amount :
10000
Enter the time period in years :
3
The simple interest is 7500.00
Would you like to continue(y/n) :
y
Interest Menu
1. Simple Interest
2. Compound Interest
Enter your choice(1/2) :
2
Enter the principle amount :
10000
Enter the time period in years :
3
Enter the no. of times interest applied per year :
3
The compound interest is 10552.23
Would you like to continue(y/n) :
n

E:\Java Programs>_
```

RESULT:

The required program was executed successfully.

2. Write an interface to find area and perimeter. Using interface find area and perimeter of square and circle.

AIM:

To write a JAVA program with an interface to find area and perimeter.

ALGORITHM:

Step 1: Start.

Step 2: Display the list of operations and get the choice from the user.

Step 3: If the choice is 1, get the side of the square and calculate its area and perimeter.

Step 4: If the choice is 2, get the radius of the circle and calculate its area and circumference.

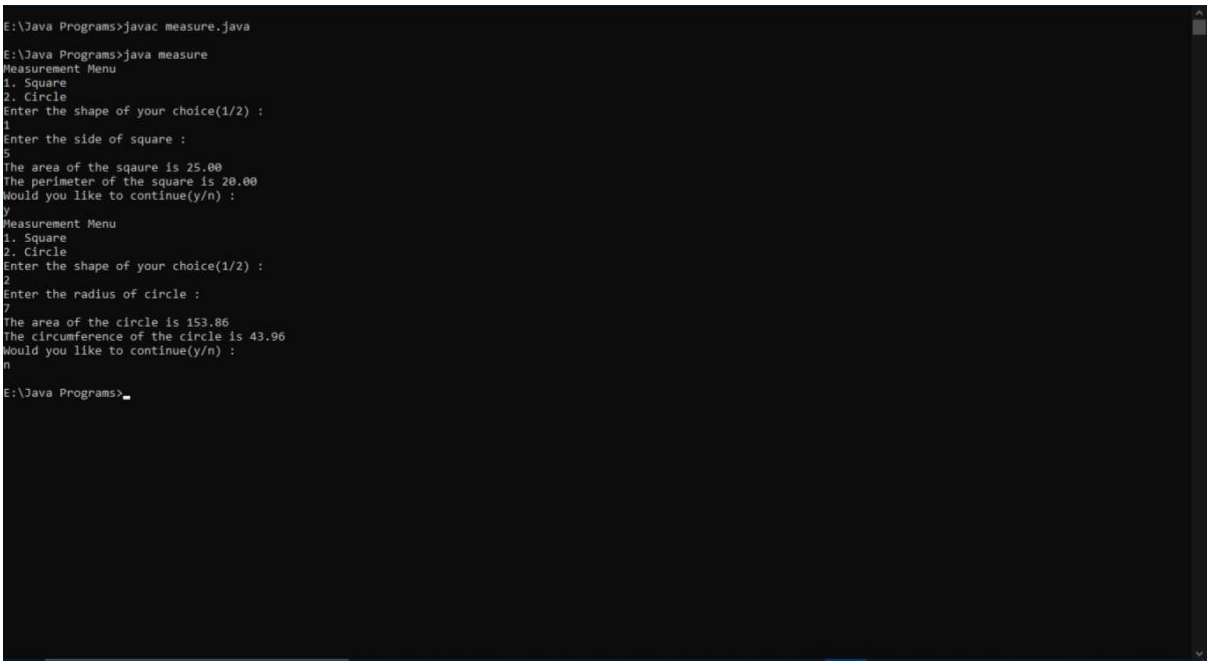
Step 5: Repeat steps 2-4, if the user wants to continue.

Step 6: End.

PROGRAM:

```
import java.util.Scanner;
interface aandp
{
    public double area();
    public double perimeter();
}
class square implements aandp
{
    private double side;
    square(double side)
    {
        this.side=side;
    }
    public double area()
    {
        return Math.pow(side,2.0);
    }
    public double perimeter()
    {
        return 4*side;
    }
}
class circle implements aandp
{
    private double radius;
    circle(double radius)
    {
        this.radius=radius;
    }
    public double area()
    {
        return 3.14*Math.pow(radius,2.0);
    }
    public double perimeter()
    {
        return 2*3.14*radius;
    }
}
public class measure
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        char  c='y';
        while(c=='y'||c=='Y')
        {
            System.out.println("Measurement Menu");
            System.out.println("1. Square");
            System.out.println("2. Circle");
            System.out.println("Enter the shape of your choice(1/2) : ");
            int ch=s.nextInt();
            if(ch==1)
            {
                System.out.println("Enter the side of square : ");
                double side=s.nextDouble();
                square sq=new square(side);
                System.out.printf("The area of the sqaure is %.2f\n",sq.area()
);
                System.out.printf("The perimeter of the square is %.2f\n",sq.p
erimeter());
            }

            else if(ch==2)
            {
                System.out.println("Enter the radius of circle : ");
                double radius=s.nextDouble();
                circle ci=new circle(radius);
                System.out.printf("The area of the circle is %.2f\n",ci.area()
);
                System.out.printf("The circumference of the circle is %.2f\n",
ci.perimeter());
            }
            else
                System.out.println("Invalid command!");
            s.nextLine();
            System.out.println("Would you like to continue(y/n) : ");
            c=s.nextLine().charAt(0);
        }
        s.close();
    }
}
```

OUTPUT:



```
E:\Java Programs>javac measure.java

E:\Java Programs>java measure
Measurement Menu
1. Square
2. Circle
Enter the shape of your choice(1/2) :
1
Enter the side of square :
5
The area of the sqaure is 25.00
The perimeter of the square is 20.00
Would you like to continue(y/n) :
y
Measurement Menu
1. Square
2. Circle
Enter the shape of your choice(1/2) :
2
Enter the radius of circle :
7
The area of the circle is 153.86
The circumference of the circle is 43.96
Would you like to continue(y/n) :
n

E:\Java Programs>_
```

RESULT:

The required program was executed successfully.

3. Write a program to create interface named customer. In this keep the methods called information(),show() and also maintain in the Tax rate. Implement this interface in employee class and calculate the tax of the employee based on their Income.

AIM:

To write a JAVA program with a interface customer.

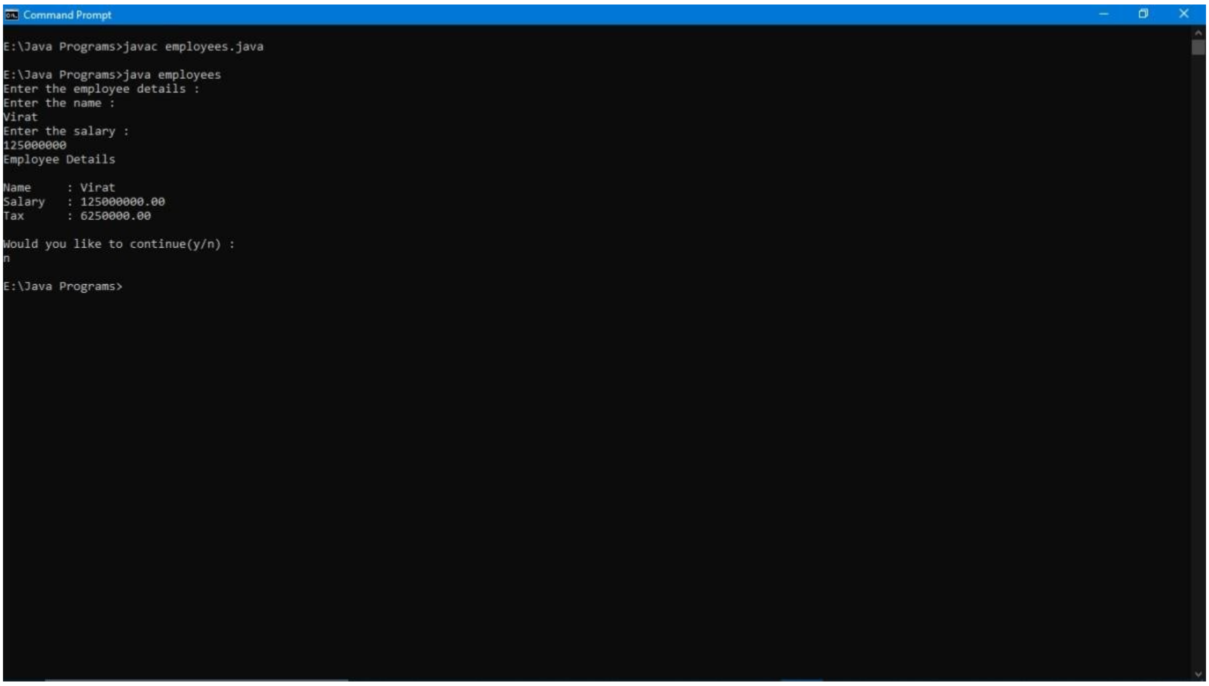
ALGORITHM:

Step 1: Start.
Step 2: Get the employee details from the user.
Step 3: Display the employee details along with the tax he/she has to pay.
Step 4: Repeat the steps 2-3, if the user wants to continue.
Step 5: End.

PROGRAM:

```java
import java.util.Scanner;
interface customer
{
    public double information();
    public void show();
    double trate=5.0;
}
public class employees implements customer
{
    private String name;
    private double sal,tax;
    employees(String name,double sal)
    {
        this.name=name;
        this.sal=sal;
        this.tax=information();
    }
    public double information()
    {
        return (sal*tax)/100.0;
    }
    public void show()
    {
        System.out.println("Employee Details\n");
        System.out.println("Name     : "+this.name);
        System.out.printf("Salary   : %.2f\n",this.sal);
        System.out.printf("Tax      : %.2f\n\n",this.tax);
    }
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        char c='y';
        while(c=='y'||c=='Y')
        {
            System.out.println("Enter the employee details : ");
            System.out.println("Enter the name : ");
            String name=s.nextLine();
            System.out.println("Enter the salary : ");
            double sal=s.nextDouble();
            s.nextLine();
            employees e=new employees(name,sal);
            e.show();
            System.out.println("Would you like to continue(y/n) :");
            c=s.nextLine().charAt(0);
        }
        s.close();
    }
}
```

OUTPUT:



```
E:\Java Programs>javac employees.java

E:\Java Programs>java employees
Enter the employee details :
Enter the name :
Virat
Enter the salary :
125000000
Employee Details

Name     : Virat
Salary   : 125000000.00
Tax      : 6250000.00

Would you like to continue(y/n) :
n

E:\Java Programs>
```

RESULT:

The required JAVA program was executed successfully.

Exp No.: 5
Date:

Construct user defined packages

Design a package for complex number addition and subtraction.

Aim:

To design a package for complex number addition and subtraction.

ALGORITHM:

a) Package:

Step 1: Start.
Step 2: Create a package ComplexNumbers, with a class ComplexNumbers.
Step 3: Define a method add(), to perform addition of complex numbers.
Step 4: Define a method sub(), to perform subtraction of complex numbers.
Step 5: End.

b) Implementation:

Step 1: Start.
Step 2: Import required packages.
Step 3: Get the two complex numbers from the user.
Step 4: Display the list of operations and get the user's choice.
Step 5: If the choice is 1, perform the addition for the complex numbers.
Step 6: If the choice is 2, perform the subtraction for the complex numbers.
Step 7: Repeat steps 3-6, if the user wants to continue.
Step 8: End.

PROGRAM:

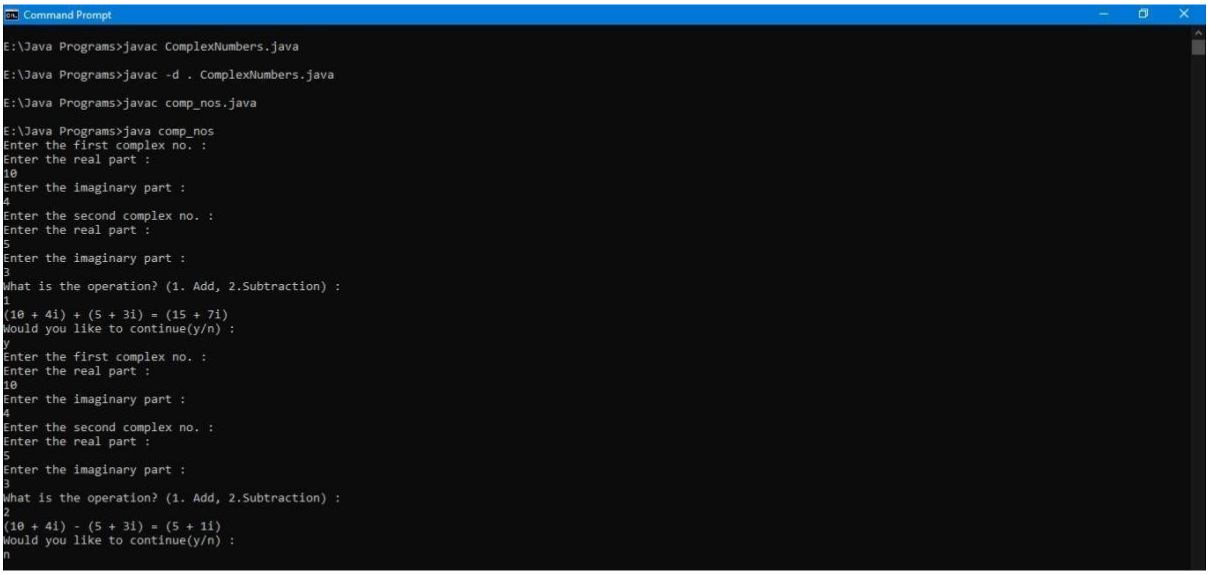
a) Package:

```
package ComplexNumbers;
public class ComplexNumbers
{
    private int real,image;
    public ComplexNumbers()
    {    }
    public ComplexNumbers(int real,int image)
    {
        this.real=real;
        this.image=image;
    }
    public static ComplexNumbers add(ComplexNumbers a,ComplexNumbers b)
    {
        ComplexNumbers sum=new ComplexNumbers();
        sum.real=a.real+b.real;
        sum.image=a.image+b.image;
        return sum;
    }
    public static ComplexNumbers sub(ComplexNumbers a,ComplexNumbers b)
    {
        ComplexNumbers diff=new ComplexNumbers();
        diff.real=a.real-b.real;
        diff.image=a.image-b.image;
        return diff;
    }
    public String toString()
    {
        if(this.image<0)
            return String.format("(%d - %di)",this.real,Math.abs(this.image));
        else
            return String.format("(%d + %di)",this.real,this.image);
    }
}
```

Implementation:

```
import java.util.Scanner;
import ComplexNumbers.ComplexNumbers;
public class comp_nos
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        char c;
        do
        {
            System.out.println("Enter the first complex no. : ");
            System.out.println("Enter the real part : ");
            int real1=s.nextInt();
            System.out.println("Enter the imaginary part : ");
            int image1=s.nextInt();
            ComplexNumbers op1=new ComplexNumbers(real1,image1);
            System.out.println("Enter the second complex no. : ");
            System.out.println("Enter the real part : ");
            int real2=s.nextInt();
            System.out.println("Enter the imaginary part : ");
            int image2=s.nextInt();
            ComplexNumbers op2=new ComplexNumbers(real2,image2);
            System.out.println("What is the operation? (1. Add, 2.Subtraction)
 : ");
            int ch=s.nextInt();
            s.nextLine();
            switch (ch)
            {
                case 1: ComplexNumbers sum=ComplexNumbers.add(op1,op2);
                        System.out.println(op1+" + "+op2+" = "+sum);
                        break;
                case 2: ComplexNumbers diff=ComplexNumbers.sub(op1,op2);
                        System.out.println(op1+" - "+op2+" = "+diff);
                        break;
                default:System.out.println("Invalid Command!");
            }
            System.out.println("Would you like to continue(y/n) : ");
            c=s.nextLine().charAt(0);
        }while(c=='y'||c=='Y');
        s.close();
    }
}
```

OUTPUT:



```
E:\Java Programs>javac ComplexNumbers.java

E:\Java Programs>javac -d . ComplexNumbers.java

E:\Java Programs>javac comp_nos.java

E:\Java Programs>java comp_nos
Enter the first complex no. :
Enter the real part :
10
Enter the imaginary part :
4
Enter the second complex no. :
Enter the real part :
5
Enter the imaginary part :
3
What is the operation? (1. Add, 2.Subtraction) :
1
(10 + 4i) + (5 + 3i) = (15 + 7i)
Would you like to continue(y/n) :
y
Enter the first complex no. :
Enter the real part :
10
Enter the imaginary part :
4
Enter the second complex no. :
Enter the real part :
5
Enter the imaginary part :
3
What is the operation? (1. Add, 2.Subtraction) :
2
(10 + 4i) - (5 + 3i) = (5 + 1i)
Would you like to continue(y/n) :
n
```

RESULT:

The required JAVA program was implemented and executed successfully.

Exp No.: 6
Date

Programs using exception handling mechanisms

1. Write a java program with exceptional handler which handles three exceptions.

AIM:

To write a JAVA program with exceptional handler which handles three exception.

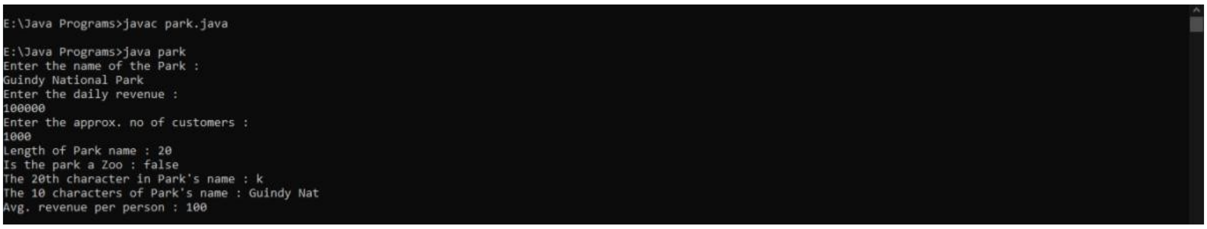
ALGORITHM:

Step 1: Start.
Step 2: Import required packages.
Step 3: Get the name of the park from the admin.
Step 4: Get the average daily revenue and customers.
Step 5: Details such as the length of the park’s name, if the park is zoo, 20th character in the park’s name, first 10 characters of the park’s name and average contribution by a customer to revenue is displayed.
Step 6: In case of any discrepancy in data, corresponding message is displayed.
Step 7: End.

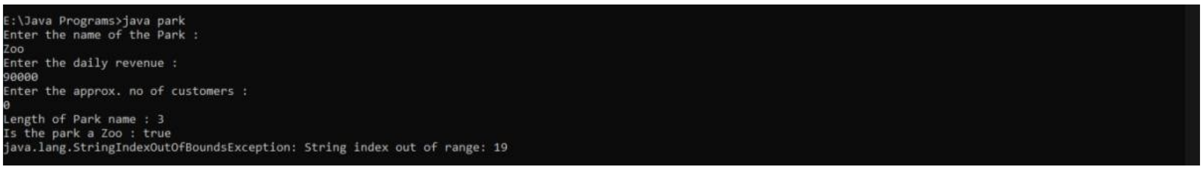
PROGRAM:

```
import java.util.Scanner;
public class park
{
    public static void main(String[] args)
    {
        Scanner s=new Scanner(System.in);
        try
        {
            System.out.println("Enter the name of the Park : ");
            String pname=s.nextLine();
            System.out.println("Enter the daily revenue : ");
            int rev=s.nextInt();
            System.out.println("Enter the approx. no of customers per day : ")
;
            int n=s.nextInt();
            s.nextLine();
            System.out.println("Length of Park name : "+pname.length());
            System.out.println("Is the park a Zoo : "+pname.contains("Zoo"));
            System.out.println("The 20th character in Park's name : "+pname.ch
arAt(19));
            System.out.println("The 10 characters of Park's name : "+pname.sub
string(0,10));
            System.out.println("Avg. revenue per person : "+(rev/n));
        }
        catch(ArithmeticException a)
        {
            System.out.println(a);
        }
        catch(NullPointerException n)
        {
            System.out.println(n);
        }
        catch(StringIndexOutOfBoundsException st)
        {
            System.out.println(st);
        }
        s.close();
    }
}
```

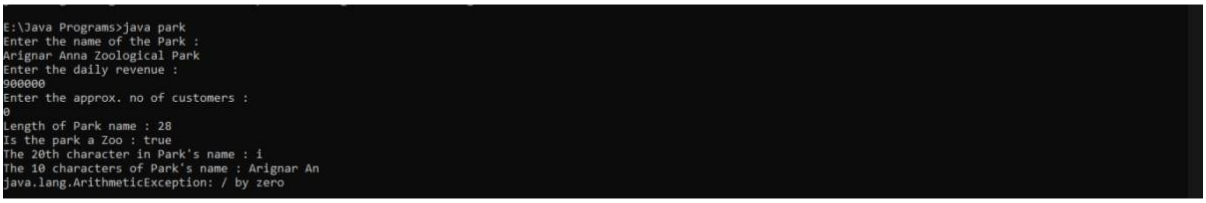
OUTPUT:



```
E:\Java Programs>javac park.java

E:\Java Programs>java park
Enter the name of the Park :
Guindy National Park
Enter the daily revenue :
100000
Enter the approx. no of customers :
1000
Length of Park name : 20
Is the park a Zoo : false
The 20th character in Park's name : k
The 10 characters of Park's name : Guindy Nat
Avg. revenue per person : 100
```



```
E:\Java Programs>java park
Enter the name of the Park :
Zoo
Enter the daily revenue :
90000
Enter the approx. no of customers :
0
Length of Park name : 3
Is the park a Zoo : true
java.lang.StringIndexOutOfBoundsException: String index out of range: 19
```



```
E:\Java Programs>java park
Enter the name of the Park :
Arignar Anna Zoological Park
Enter the daily revenue :
900000
Enter the approx. no of customers :
0
Length of Park name : 28
Is the park a Zoo : true
The 20th character in Park's name : i
The 10 characters of Park's name : Arignar An
java.lang.ArithmeticException: / by zero
```

RESULT:

The required JAVA program was implemented and executed successfully.

2. Write a java program to input age from user and throw user-defined exception if entered age is negative or age <18.

AIM:

To write a JAVA program to get input from the user and throw an user-defined exception, if the age is negative or age<18.

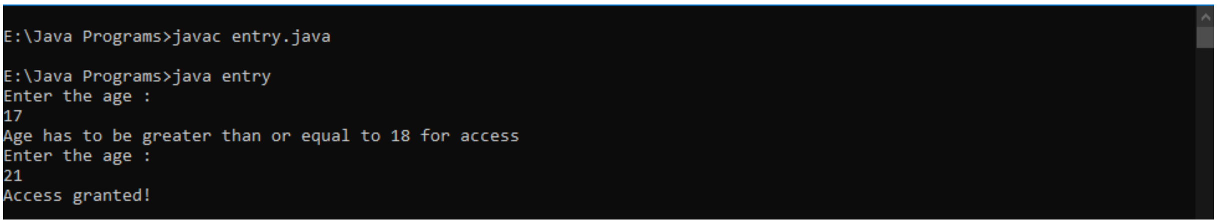
ALGORITHM:

```
Step 1: Start.
Step 2: Import required packages.
Step 3: Get the age of the user.
Step 4: If the age is greater than or equal to 18, the user is granted entry acces.
Step 5: Else an exception is thrown prompting the minimum age for entry.
Step 6: Repeat steps 3-5, until an user is granted access.
Step 7: End.
```

PROGRAM:

```
import java.util.Scanner;
class AgeLessThanException extends Exception
{
    public String toString()
    {
        return "Age has to be greater than or equal to 18 for access";
    }
}
public class entry
{
    public static void main(String[] args)
    {
        Scanner s=new Scanner(System.in);
        while(true)
        {
            try
            {
                System.out.println("Enter the age : ");
                int age=s.nextInt();
                if(age<18)
                {
                    throw new AgeLessThanException();
                }
                else
                {
                    System.out.println("Access granted!");
                    break;
                }
            }
            catch(AgeLessThanException e)
            {
                System.out.println(e);
                continue;
            }
        }
        s.close();
    }
}
```

OUTPUT:



```
E:\Java Programs>javac entry.java

E:\Java Programs>java entry
Enter the age :
17
Age has to be greater than or equal to 18 for access
Enter the age :
21
Access granted!
```

RESULT:

The required JAVA program was implemented and executed successfully.

3. Write a java program to register the students when their age is less than 12 and weight is less than 40, if any of the condition is not met then the user should get an ArithmeticException with the warning message “Student is not eligible for registration”.

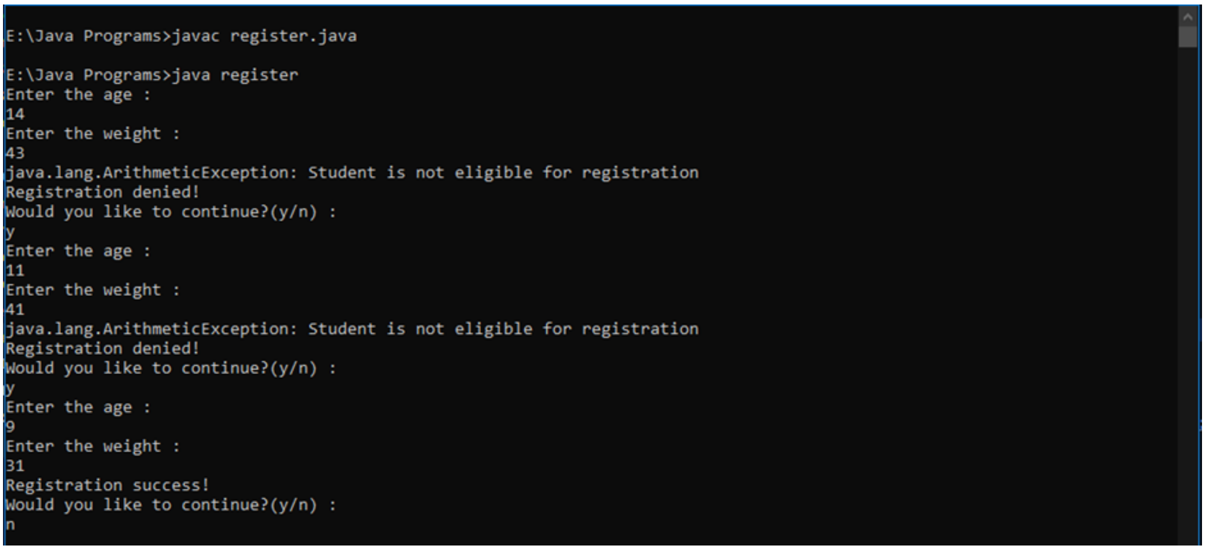
AIM:
To write a JAVA program to register the students when their age is less than 12 and weight is less than 40, throw an ArithmeticException if any of the conditions is not meant.

ALGORITHM:

```
Step 1: Start.
Step 2: Import required packages.
Step 3: Get the age and weight of the student from user.
Step 4: If the age is less than 12 and weight is less than 40, the student can register himself/herself.
Step 5: Else, an ArithmeticException with a prompt “Student is not eligible for registration”, is thrown.
Step 6: Repeat steps 3-5, if the user wants to continue. Step 7: End.
```

PROGRAM:

```
import java.util.Scanner;
public class register
{
public static void main(String[] args)
{
Scanner s=new Scanner(System.in);
char c;
do
{

try
{
System.out.println("Enter the age : ");
int age=s.nextInt();
System.out.println("Enter the weight : ");
 int wei=s.nextInt();
if(age<12 && wei<40)
System.out.println("Registration success!");
Else
    throw new ArithmeticException("Student is not eligible for registration");
    }


catch(ArithmeticException a)
{
System.out.println(a); System.out.println("Registration denied!");
}
System.out.println("Would you like to continue?(y/n) : ");
c=s.nextLine().charAt(0);
}while(c=='y'||c=='Y'); s.close();
}
}
```



```
E:\Java Programs>javac register.java

E:\Java Programs>java register
Enter the age :
14
Enter the weight :
43
java.lang.ArithmeticException: Student is not eligible for registration
Registration denied!
Would you like to continue?(y/n) :
y
Enter the age :
11
Enter the weight :
41
java.lang.ArithmeticException: Student is not eligible for registration
Registration denied!
Would you like to continue?(y/n) :
y
Enter the age :
9
Enter the weight :
31
Registration success!
Would you like to continue?(y/n) :
n
```

RESULT:

The required JAVA program was implemented and executed successfully

Exp No.: 7
Date:

Develop programs using multithreaded programs

1. Write a java program to print different multiplication tables using multiple threads.

AIM:

To write a JAVA program to print different multiplication tables using multiple threads.

ALGORITHM:

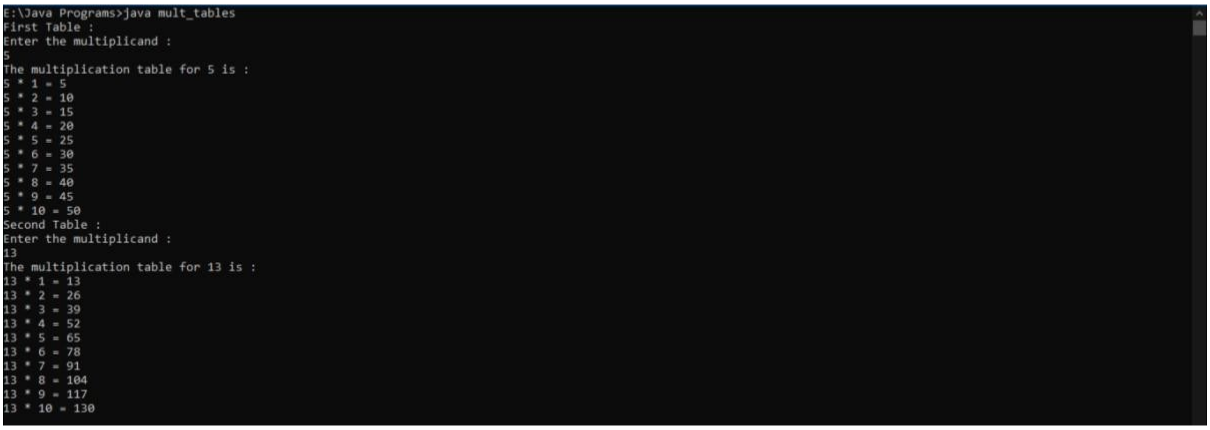
```
Step 1: Start.
Step 2: Get the multiplicand of the first table from the user.
Step 3: Print the first multiplication table.
Step 4: Get the multiplicand of the second table from the user.
Step 5: Print the second multiplication table.
Step 6: End.
```

PROGRAM:

```
import java.util.Scanner;
class table extends Thread
{
    public void run()
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the multiplicand : ");
        int m=s.nextInt();
        System.out.println("The multiplication table for "+m+" is : ");
        for(int i=1;i<=10;i++)
            System.out.printf("%d * %d = %d\n",m,i,m*i);
    }
}
public class mult_tables
{
    public static void main(String[] args)
    {
        table t1=new table();
        table t2=new table();
        try
        {
            System.out.println("First Table : ");
            t1.start();
            t1.join();
            System.out.println("Second Table : ");
            t2.start();
            t2.join();
        }
```

```
             catch (Exception e)
        {
            System.out.println(e);
        }
    }
}
```

OUTPUT:



```
E:\Java Programs>java mult_tables
First Table :
Enter the multiplicand :
5
The multiplication table for 5 is :
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
Second Table :
Enter the multiplicand :
13
The multiplication table for 13 is :
13 * 1 = 13
13 * 2 = 26
13 * 3 = 39
13 * 4 = 52
13 * 5 = 65
13 * 6 = 78
13 * 7 = 91
13 * 8 = 104
13 * 9 = 117
13 * 10 = 130
```

RESULT:

The required JAVA program was implemented and executed successfully.

2. Create a bank database application program to illustrate the use of multithread.

AIM:

To create a bank database application program to illustrate the use of multithread.

ALGORITHM:

Step 1: Start.

Step 2: Import required packages.

Step 3: Get the current account balance.

Step 4: Display the list of operations to user and get his/her choice.

Step 5: If the choice is 1, get the amount of withdrawal from the user.

Step 5i: If the amount is less than or equal to the current balance, the money is dispensed and balance is updated.

Step 5ii: Else, the user is prompted that there isn’t sufficient balance for the withdrawal.

Step 6: If the choice is 2, get the amount of deposit from the user and update the balance.

Step 7: Repeat steps 4-6, if the user wants to continue.

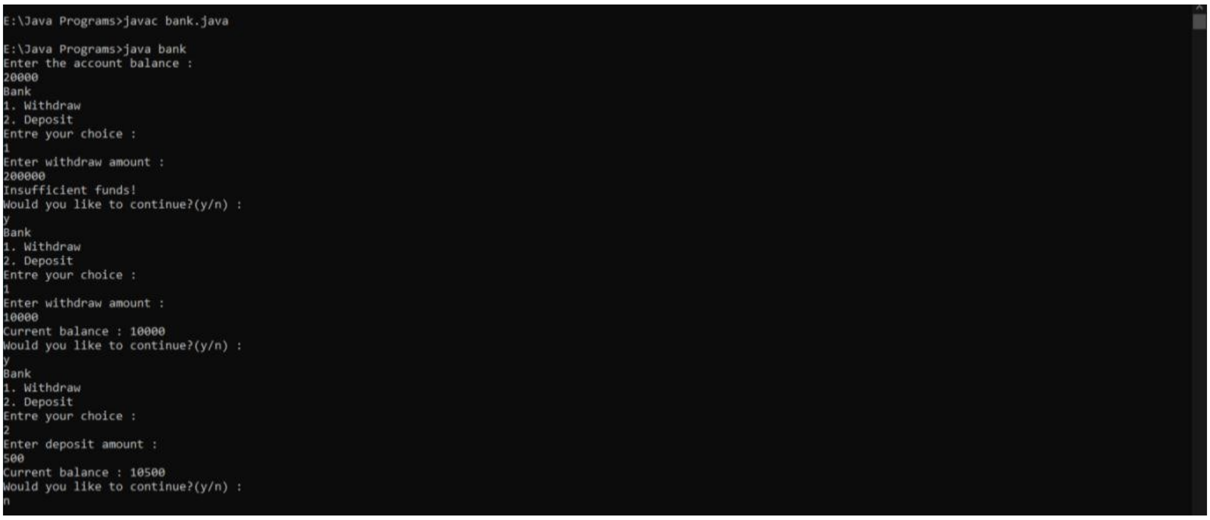
Step 8: End.

PROGRAM:

```
import java.util.Scanner;
class account
{
    int balance;
    account(int balance)
    {
        this.balance=balance;
    }
    public int getBalance()
    {
        return balance;
    }
    synchronized public void update(int amt)
    {
        this.balance+=amt;
    }
}
class withdraw extends Thread
{
    account a;
    withdraw(account a)
    {
        this.a=a;
    }
```

```
      public void run()
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter withdraw amount : ");
        int amt=s.nextInt();
        s.nextLine();
        if(amt<=a.getBalance())
        {
            a.update(-amt);
            System.out.println("Current balance : "+a.getBalance());
        }
        else
            System.out.println("Insufficient funds!");
    }
}
class deposit extends Thread
{
    account a;
    deposit(account a)
    {
        this.a=a;
    }
    public void run()
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter deposit amount : ");
        int amt=s.nextInt();
        s.nextLine();
        a.update(amt);
        System.out.println("Current balance : "+a.getBalance());
    }
}
public class bank
{
    public static void main(String[] args)
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the account balance : ");
        int balance=s.nextInt();
        account a=new account(balance);
        char c;
        do
        {
            System.out.println("Bank");
            System.out.println("1. Withdraw");
            System.out.println("2. Deposit");
            System.out.println("Entre your choice : ");
            int ch=s.nextInt();
            s.nextLine();
            try
            {
                switch(ch)
                {
                    case 1: withdraw w=new withdraw(a);
                            w.start();
                            w.join();
                            break;
                    case 2: deposit d=new deposit(a);
                            d.start();
                            d.join();
                            break;
                    default:System.out.println("Invalid command!");
                }
            }
            catch(Exception e)
            {
                System.out.println(e);
            }
            System.out.println("Would you like to continue?(y/n) : ");
            c=s.nextLine().charAt(0);
        }while(c=='y'||c=='Y');
        s.close();
    }
}
```

OUTPUT:



```
E:\Java Programs>javac bank.java

E:\Java Programs>java bank
Enter the account balance :
20000
Bank
1. Withdraw
2. Deposit
Entre your choice :
1
Enter withdraw amount :
200000
Insufficient funds!
Would you like to continue?(y/n) :
y
Bank
1. Withdraw
2. Deposit
Entre your choice :
1
Enter withdraw amount :
10000
Current balance : 10000
Would you like to continue?(y/n) :
y
Bank
1. Withdraw
2. Deposit
Entre your choice :
2
Enter deposit amount :
500
Current balance : 10500
Would you like to continue?(y/n) :
n
```

RESULT:

The required JAVA program is implemented and executed successfully.

Exp No.: 8

Execute Program using File streams.

1. Write a Java program to write and read a plain text file (use byte Streams).

AIM:

To write a JAVA program to write and read a plain text using byte streams.

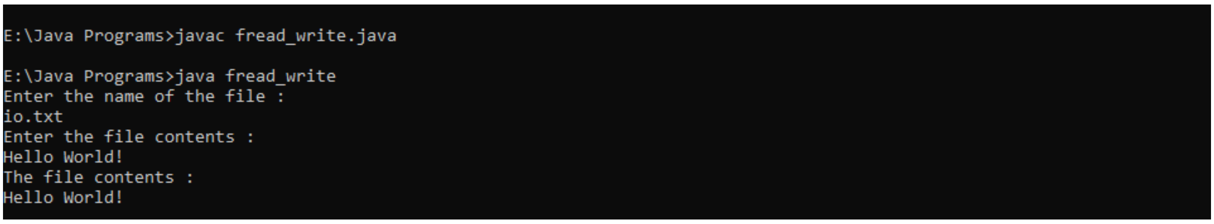
ALGORITHM:

Step 1: Start.
Step 2: Import required packages.
Step 3: Get the name of the file to be created, from the user.
Step 4: Get the text to be written on to the file.
Step 5: Create an output stream, write the text on the file and close the stream.
Step 6: Create an input stream, read the file and close the stream.
Step 7: Display the contents read from the file.
Step 8: End.

PROGRAM:

```java
import java.io.*;
import java.util.Scanner;
public class fread_write
{
    public static void main(String[] args)
    {
        FileOutputStream fout;
        FileInputStream fin;
        Scanner s=new Scanner(System.in);
        try
        {
            System.out.println("Enter the name of the file : ");
            String fname=s.nextLine();
            fout=new FileOutputStream(fname);
            System.out.println("Enter the file contents : ");
            String cont=s.nextLine();
            fout.write(cont.getBytes());
            fout.close();
            fin=new FileInputStream(fname);
            System.out.println("The file contents : ");
            byte[] out=new byte[100];
            fin.read(out);
            String data=new String(out);
            System.out.println(data);
            fin.close();
        }
        catch(IOException e)
        {
            System.out.println(e);
        }
        s.close();
    }
}
```

OUTPUT:



```
E:\Java Programs>javac fread_write.java

E:\Java Programs>java fread_write
Enter the name of the file :
io.txt
Enter the file contents :
Hello World!
The file contents :
Hello World!
```

RESULT:

The JAVA program to write and read a file using byte stream was implemented and executed successfully.

2. Write a Java program to append text to an existing file.

AIM:

To write a JAVA program to append text to an existing file.

ALGORITHM:

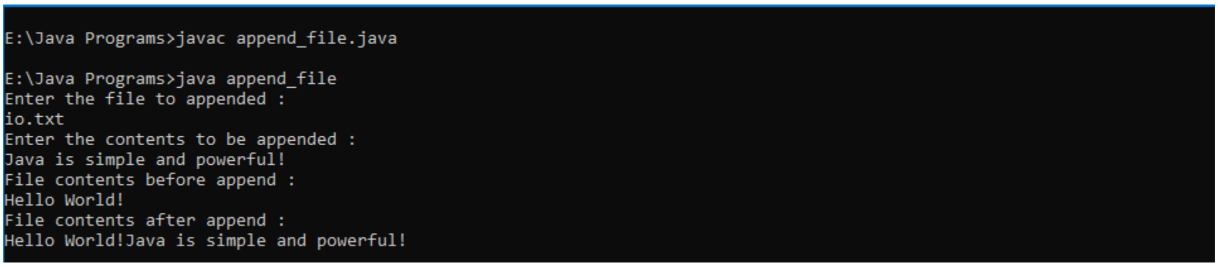
```
Step 1: Start.
Step 2: Import required packages.
Step 3: Get the name of the file to appended from the user.
Step 4: Get the text to be appended to the file.
Step 5: Display the contents of the file before appending.
Step 6: Create an output stream in append mode and append the text to the file.
Step 7: Display the final contents of the file.
Step 8: End.
```

PROGRAM:

```java
import java.io.*;
import java.util.Scanner;
public class append_file
{
    public static void main(String[] args)
    {
        Scanner s=new Scanner(System.in);
        FileOutputStream fout;
        FileInputStream fin;
        byte[] out=new byte[100];
        String data;
        try
        {
            System.out.println("Enter the file to appended : ");
            String fname=s.nextLine();
            fout=new FileOutputStream(fname,true);
            fin=new FileInputStream(fname);
            System.out.println("Enter the contents to be appended : ");
            String cont=s.nextLine();
            int off=fin.available();
            if(off>0)
            {
                fin.read(out);
                fin.skip(-off);
                data=new String(out);
                System.out.println("File contents before append : ");
                System.out.println(data);
            }
            else
                System.out.println("File doesn't exist, a new file will be cre
ated.");
```

```
            fout.write(cont.getBytes());
            fin.read(out);
            data=new String(out);
            System.out.println("File contents after append : ");
            System.out.println(data);
            fout.close();
            fin.close();
        }
        catch(IOException e)
        {
            System.out.println(e);
        }
        s.close();
    }
}
```

OUTPUT:



```
E:\Java Programs>javac append_file.java

E:\Java Programs>java append_file
Enter the file to appended :
io.txt
Enter the contents to be appended :
Java is simple and powerful!
File contents before append :
Hello World!
File contents after append :
Hello World!Java is simple and powerful!
```

RESULT:

The JAVA program to append text to an existing file was implemented and executed successfully.

3. Write a java program to show how to use BufferedReader and BufferedWriter classes for performing buffered I/O operations.

AIM:

To write a JAVA program to show how to use BufferedReader and BufferedWriter classes for performing buffered I/O operation.

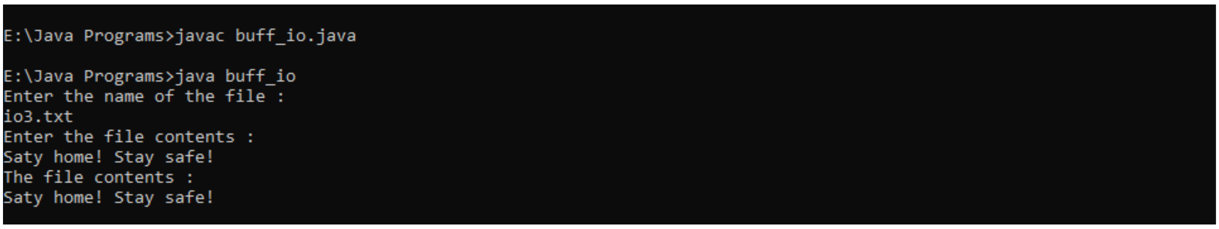
ALGORITHM:

Step 1: Start.
Step 2: Import required packages.
Step 3: Get the name of the file from the user.
Step 4: Create file streams for the file.
Step 5: Create BufferedReader and BufferedWriter on top of the file stream.
Step 6: Get the contents to be written on to the file.
Step 7: Write the contents on to the file,
Step 8: Read the file contents and display it to the user.
Step 9: End.

PROGRAM:

```java
import java.io.*;
import java.util.Scanner;
public class buff_io
{
    public static void main(String[] args)
    {
        BufferedOutputStream bout;
        BufferedInputStream bin;
        Scanner s=new Scanner(System.in);
        try
        {
            System.out.println("Enter the name of the file : ");
            String fname=s.nextLine();
            FileOutputStream fout=new FileOutputStream(fname);
            bout=new BufferedOutputStream(fout);
            System.out.println("Enter the file contents : ");
            String cont=s.nextLine();
            bout.write(cont.getBytes());
            bout.close();
            fout.close();
            FileInputStream fin=new FileInputStream(fname);
            bin=new BufferedInputStream(fin);
            System.out.println("The file contents : ");
            byte[] out=new byte[100];
            bin.read(out);
            String data=new String(out);
            System.out.println(data);
            bin.close();
            fin.close();
        }
        catch(IOException e)
        {
            System.out.println(e);
        }
        s.close();
    }
}
```

OUTPUT:



```
E:\Java Programs>javac buff_io.java

E:\Java Programs>java buff_io
Enter the name of the file :
io3.txt
Enter the file contents :
Saty home! Stay safe!
The file contents :
Saty home! Stay safe!
```

RESULT:

The JAVA program to show how to use BufferedReader and BufferedWriter was implemented and executed successfully.

Exp No.: 9

Develop Applications using Java Database Connectivity

Develop java application for employee information system (to add emp, to delete emp, to update emp, count no of emp, calculate pay).

AIM:

To develop a JAVA application for employee information system.

ALGORITHM:

```
Step 1: Start.
Step 2: Import required packages.
Step 3: Invoke the database driver and create a connection to the database.
Step 4: Display the list of all operations and get the user's choice.
Step 5: If the choice is 1, display the whole contents of the table.
Step 6: If the choice is 2, get the required data from the user and add a new record to the database.
Step 7: If the choice is 3, get the eid of the required employee and delete the record from the table.
Step 8: If the choice is 4, ask the user as to which column must be updated.
    Step 8i: If the choice is 1, list the options for salary updation and get the user's choice.
        Step 8ii: If the choice is 1, get the increment amount and increment the salary of all the employees.
        Step 8iii: If the choice is 2, get the eid and the new salary of the employee and update the salary.
    Step 8iv: If the choice 2, get the new bonus for all the employees and update it in the database.
    Step 8v: If the choice is 3, get the eid and the new role of the employee and update it in the database.
Step 9: If the choice is 5, display the total count of employees in the database.
Step 10: If the choice is 6, get the eid of the employee, calculate the pay including the bonus and print it.
Step 11: Repeat steps 4-10, if the user wants to continue.
Step 12: Close the connection to the database.
Step 13: End.
```

PROGRAM:
For accessing table in MySQL:
Table Creation:

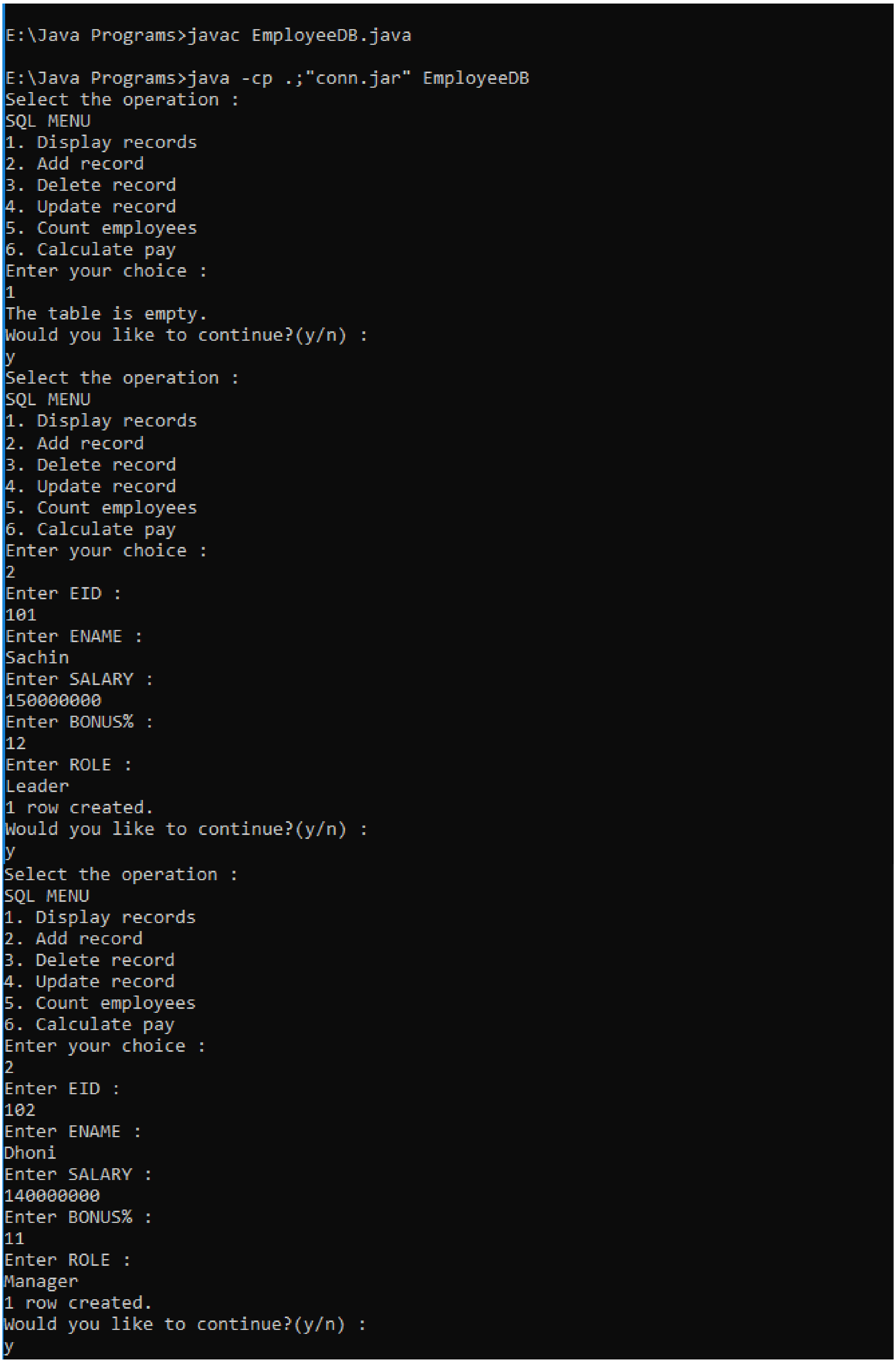
create table employee_db
(eid int,
ename varchar(20),
salary int,
bonus int,
role varchar(20));

JDBC Program:

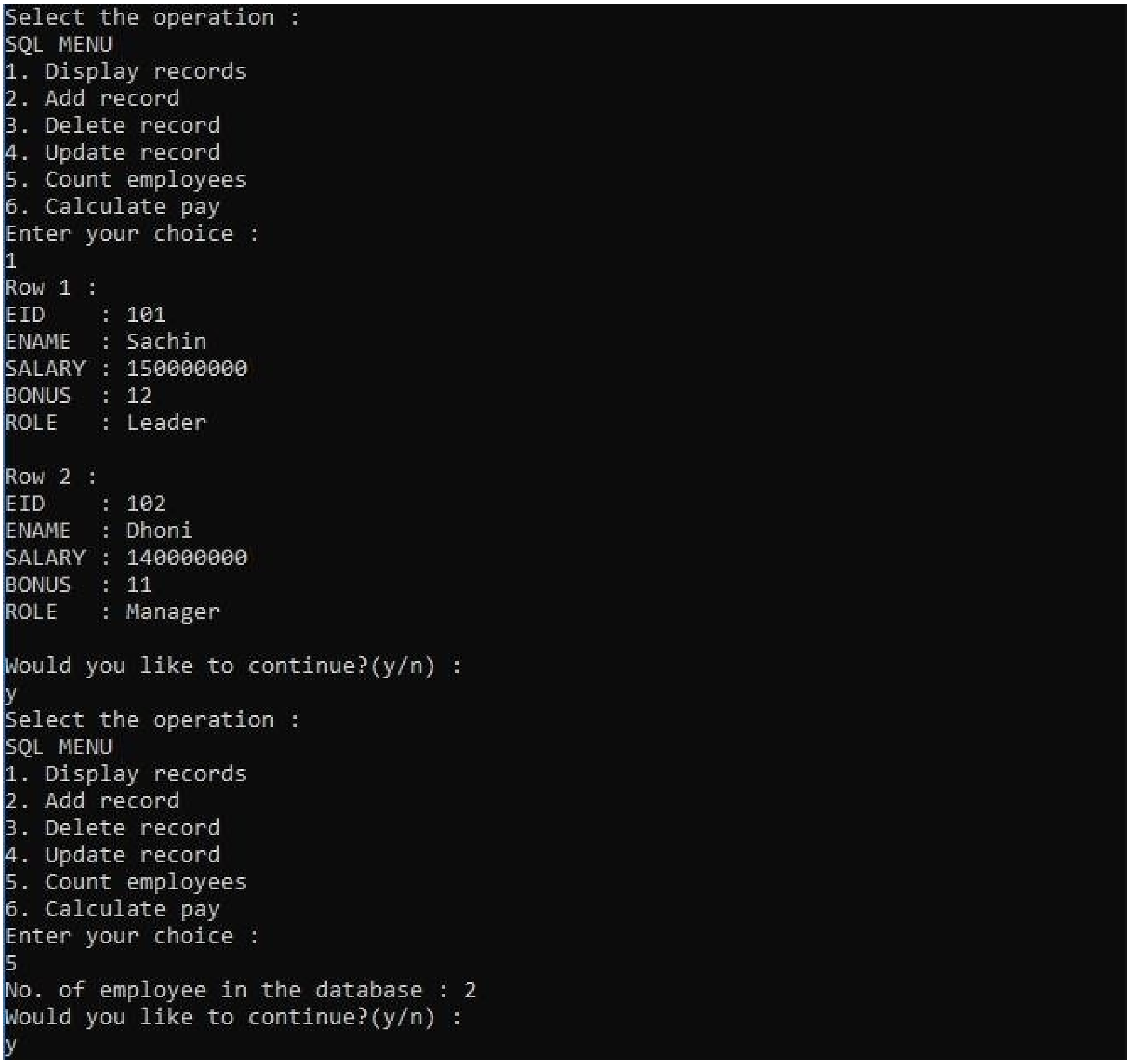
```
import  java.sql.*;
import java.util.Scanner;
public class EmployeeDB
{
    public void display(Statement stmt)
    {
        try
        {
            String q="select * from employee_db";
            ResultSet r=stmt.executeQuery(q);
            int i=0;
            if(r.next()==false)
            {
                System.out.println("The table is empty.");
                return;
            }
            do
            {
                System.out.println("Row "+(++i)+" : ");
                System.out.println("EID    : "+r.getString("EID"));
                System.out.println("ENAME : "+r.getString("ENAME"));
                System.out.println("SALARY : "+r.getString("SALARY"));
                System.out.println("BONUS : "+r.getString("BONUS"));
                System.out.println("ROLE   : "+r.getString("ROLE"));
                System.out.println();
            }while(r.next());
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
    public void add(Statement stmt,Scanner s)
    {
        try
        {
            System.out.println("Enter EID : ");
            int eid=s.nextInt();
            s.nextLine();
            System.out.println("Enter ENAME : ");
            String ename=s.nextLine();
            System.out.println("Enter SALARY : ");
            int salary=s.nextInt();
            s.nextLine();
            System.out.println("Enter BONUS% : ");
            int bonus=s.nextInt();
            s.nextLine();
            System.out.println("Enter ROLE : ");
            String role=s.nextLine();
            String q=String.format("insert into employee_db values(%d,'%s',%d,
%d,'%s')",eid,ename,salary,bonus,role);
            int status=stmt.executeUpdate(q);
            if(status==1)
                System.out.println("1 row created.");
            else
                System.out.println("Couldn't create the row!");
        }

        catch(Exception e)
        {
            System.out.println(e);
        }
    }
    public void delete(Statement stmt,Scanner s)
    {
        try
        {
            System.out.println("Enter the EID of the record to be deleted : ")
;
            int eid=s.nextInt();
            s.nextLine();
            String q=String.format("delete from employee_db where eid=%d",eid)
;
            int status=stmt.executeUpdate(q);
            if(status==1)
                System.out.println("1 row deleted.");
            else
                System.out.println("Couldn't delete the row!");
        }

        catch(Exception e)
        {
            System.out.println(e);
        }
    }
    public void update(Statement stmt,Scanner s)
    {
        try
        {
            String q;
            int status;
            System.out.println("Select the column to be updated : ");
            System.out.println("1. SALARY");
            System.out.println("2. BONUS");
            System.out.println("3. ROLE");
            System.out.println("Enter your choice : ");
            int ch=s.nextInt();
            s.nextLine();
            if(ch==1)
            {
                System.out.println("SALARY UPDATE MENU");
                System.out.println("1. Increment salary for everyone");
                System.out.println("2. Update a new salary for an employee");
                System.out.println("Enter your choice : ");
                int c=s.nextInt();
                s.nextLine();
                if(c==1)
                {
                    System.out.println("Enter the increment amount : ");
                    int incr=s.nextInt();
                    s.nextLine();
                    q=String.format("update employee_db set salary=salary+%d",
incr);

                    status=stmt.executeUpdate(q);
                    if(status==1)
                        System.out.println("1 row updated");
                    else if(status>1)
                        System.out.println(status+" rows updated.");
                    else
                        System.out.println("Couldn't update!");
                }

                else if(c==2)
                {
                    System.out.println("Enter the EID of the employee : ");
                    int eid=s.nextInt();
                    s.nextLine();
                    System.out.println("Enter the new salary : ");
                    int sal=s.nextInt();
                    s.nextLine();
                    q=String.format("update employee_db set SALARY=%d where EI
D=%d",sal,eid);

                    status=stmt.executeUpdate(q);
                    if(status!=0)
            System.out.println("1 row updated.");
          else
            System.out.println("Couldn't update!");
        }


        else
          System.out.println("Invalid Command!");
      }
      else if(ch==2)
      {
        System.out.println("Enter the new bonus for all employees : ")
;
        int b=s.nextInt();
        s.nextLine();
        q=String.format("update employee_db set bonus=%d",b);
        status=stmt.executeUpdate(q);
        if(status==1)
          System.out.println("1 row updated.");
        else if(status>1)
          System.out.println(status+" rows updated.");
        else
          System.out.println("Couldn't update!");
      }
      else if(ch==3)
      {
        System.out.println("Enter the EID of the employee : ");
        int eid=s.nextInt();
        s.nextLine();
        System.out.println("Enter the new ROLE : ");
        String role=s.nextLine();
        q=String.format("update employee_db set role='%s' where eid=%d
",role,eid);
        status=stmt.executeUpdate(q);
        if(status!=0)
          System.out.println(status+" row updated");
        else
          System.out.println("Couldn't update!");
      }
      else
        System.out.println("Invalid Command!");
    }

    catch(Exception e)
    {
      System.out.println(e);
    }
  }
  public void count(Statement stmt)
  {
    try
```

```
        {
            String q="select count(*) from employee_db";
            ResultSet r=stmt.executeQuery(q);
            r.next();
            System.out.println("No. of employee in the database : "+r.getStrin
g("COUNT(*)"));
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
    public void pay(Statement stmt,Scanner s)
    {
        try
        {
            System.out.println("Enter the EID of the employee : ");
            int eid=s.nextInt();
            s.nextLine();
            String q=String.format("select salary,bonus from employee_db where
 eid=%d",eid);
            ResultSet r=stmt.executeQuery(q);
            r.next();
            int  sal=r.getInt("SALARY");
            int bon=r.getInt("BONUS");
            double pay=(1+(bon/100.0))*sal;
            System.out.println("Employee's pay : "+pay);
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
    public static void main(String[] args)
    {
        Scanner s=new Scanner(System.in);
        try
        {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con=DriverManager.getConnection("jdbc:mysql://localhost
:3306/db","root","root");
            char c;
            do
            {
                System.out.println("Select the operation : ");
                System.out.println("SQL MENU");
                System.out.println("1. Display records");
                System.out.println("2. Add record");
                System.out.println("3. Delete record");
                System.out.println("4. Update record");
                System.out.println("5. Count employees");
                System.out.println("6. Calculate pay");
                System.out.println("Enter your choice : ");
                int ch=s.nextInt();
                s.nextLine();
                EmployeeDB e=new EmployeeDB();
                Statement stmt=con.createStatement();
                switch(ch)
                {
                    case 1: e.display(stmt);
                            break;
                    case 2: e.add(stmt,s);
                            break;
                    case 3: e.delete(stmt,s);
                            break;
                    case 4: e.update(stmt,s);
                            break;
                    case 5: e.count(stmt);
                            break;
                    case 6: e.pay(stmt,s);
                            break;
                    default:System.out.println("Invalid Command!");
                }
                System.out.println("Would you like to continue?(y/n) : ");
                c=s.nextLine().charAt(0);
            }while(c=='y'||c=='Y');
            con.close();
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        s.close();
    }
}
```

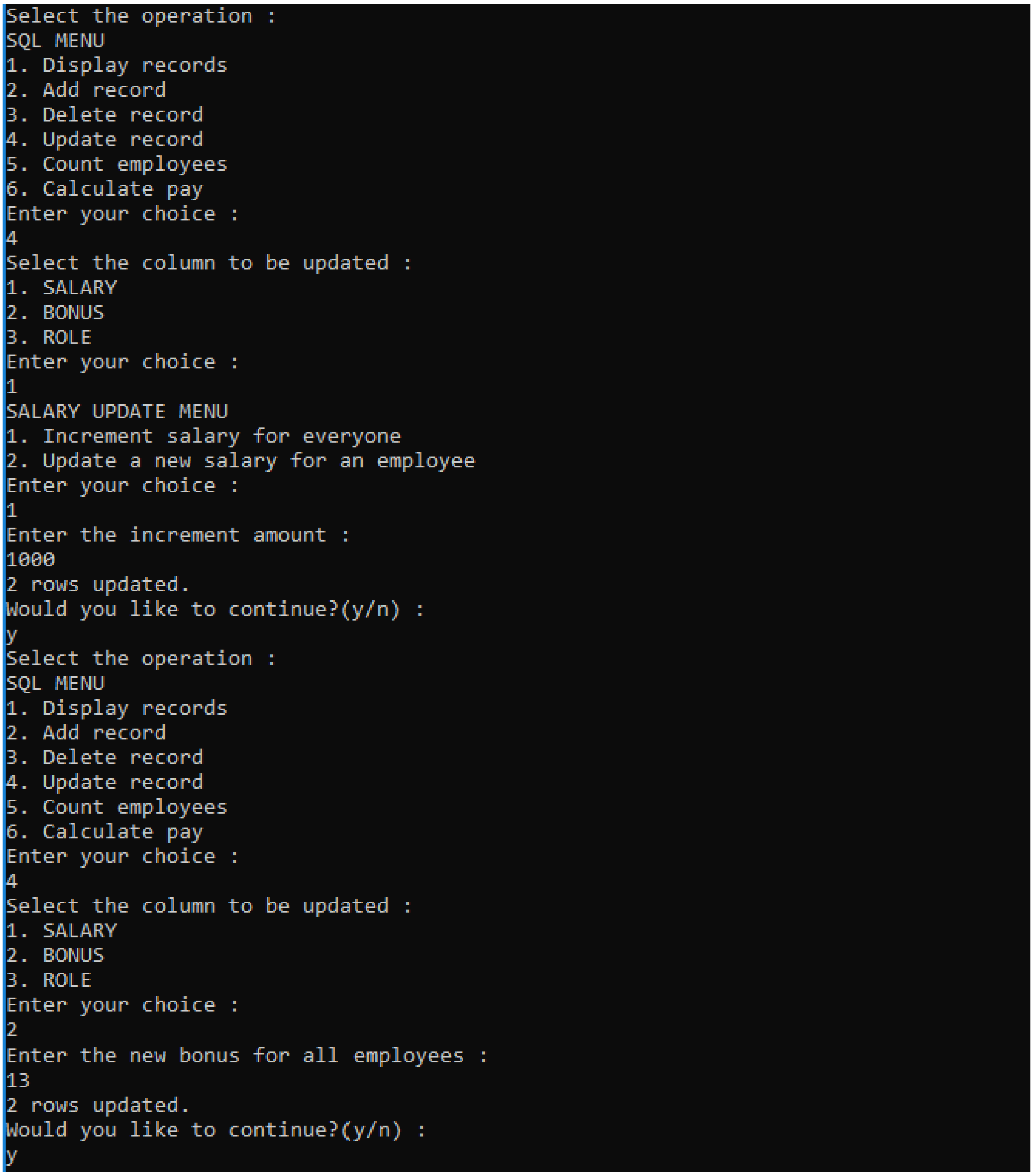
OUTPUT:



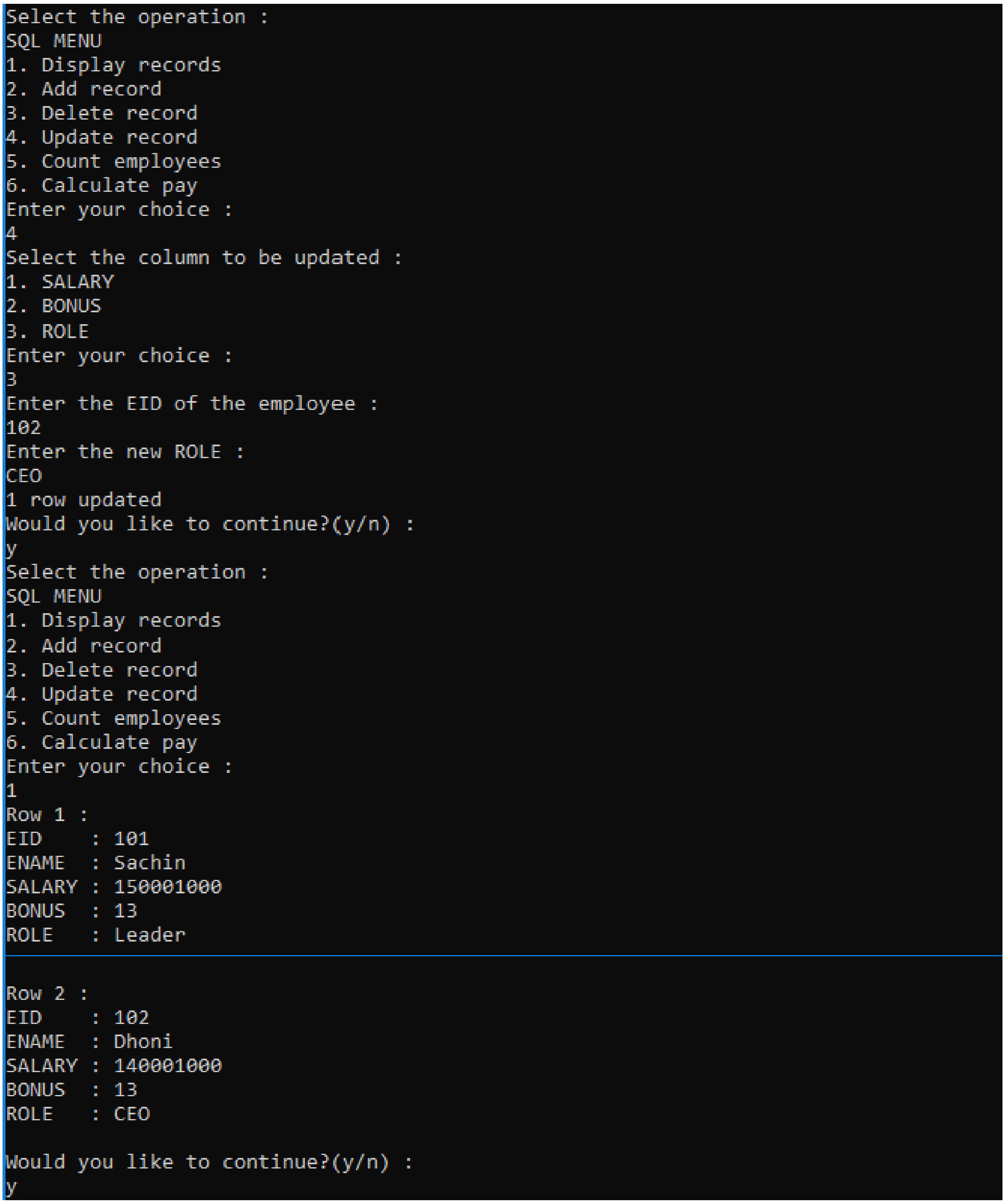
```
E:\Java Programs>javac EmployeeDB.java

E:\Java Programs>java -cp .;"conn.jar" EmployeeDB
Select the operation :
SQL MENU
1. Display records
2. Add record
3. Delete record
4. Update record
5. Count employees
6. Calculate pay
Enter your choice :
1
The table is empty.
Would you like to continue?(y/n) :
y
Select the operation :
SQL MENU
1. Display records
2. Add record
3. Delete record
4. Update record
5. Count employees
6. Calculate pay
Enter your choice :
2
Enter EID :
101
Enter ENAME :
Sachin
Enter SALARY :
150000000
Enter BONUS% :
12
Enter ROLE :
Leader
1 row created.
Would you like to continue?(y/n) :
y
Select the operation :
SQL MENU
1. Display records
2. Add record
3. Delete record
4. Update record
5. Count employees
6. Calculate pay
Enter your choice :
2
Enter EID :
102
Enter ENAME :
Dhoni
Enter SALARY :
140000000
Enter BONUS% :
11
Enter ROLE :
Manager
1 row created.
Would you like to continue?(y/n) :
y
```



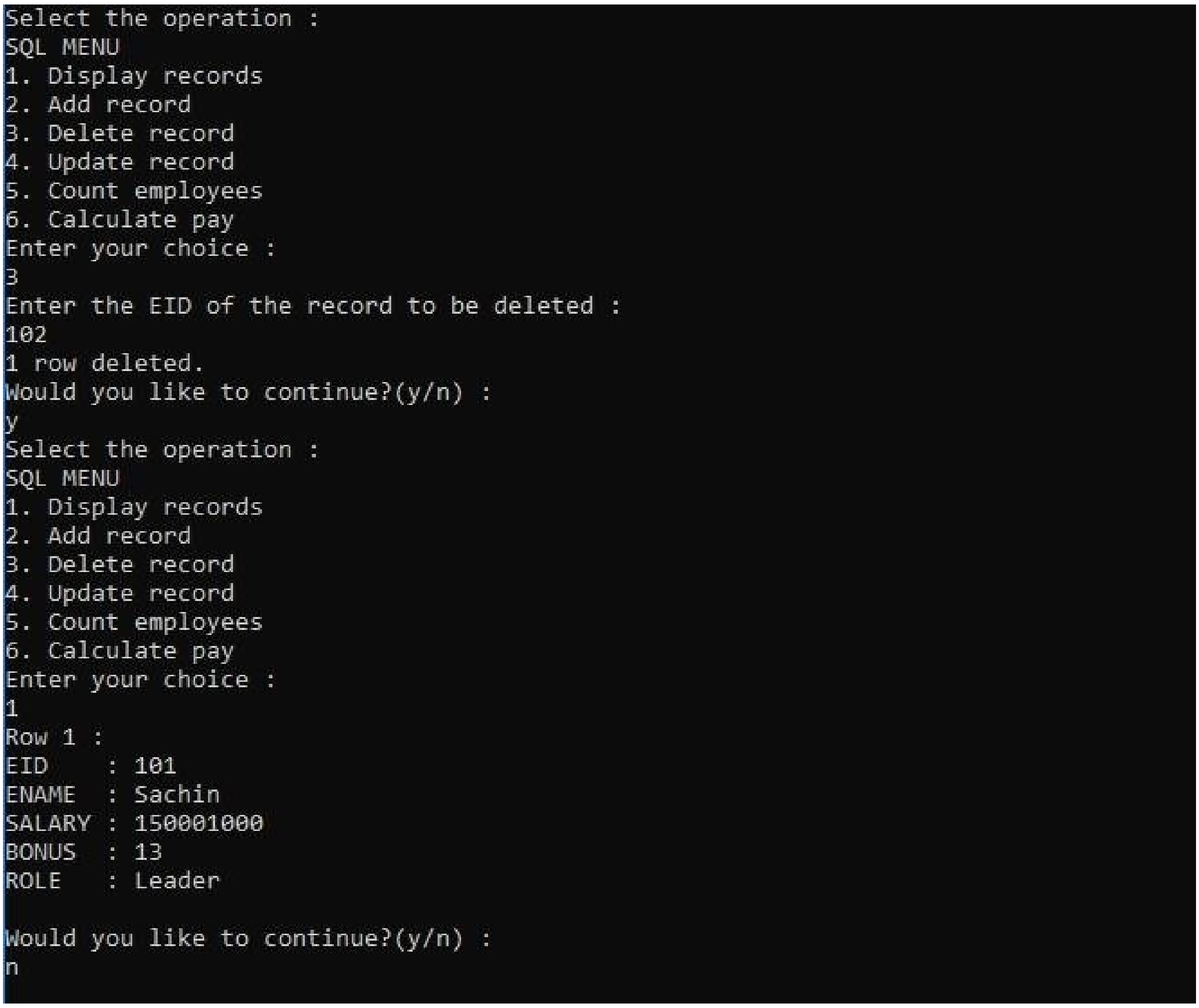
```
Select the operation :
SQL MENU
1. Display records
2. Add record
3. Delete record
4. Update record
5. Count employees
6. Calculate pay
Enter your choice :
1
Row 1 :
EID    : 101
ENAME  : Sachin
SALARY : 150000000
BONUS  : 12
ROLE   : Leader

Row 2 :
EID    : 102
ENAME  : Dhoni
SALARY : 140000000
BONUS  : 11
ROLE   : Manager

Would you like to continue?(y/n) :
y
Select the operation :
SQL MENU
1. Display records
2. Add record
3. Delete record
4. Update record
5. Count employees
6. Calculate pay
Enter your choice :
5
No. of employee in the database : 2
Would you like to continue?(y/n) :
y
```



```
Select the operation :
SQL MENU
1. Display records
2. Add record
3. Delete record
4. Update record
5. Count employees
6. Calculate pay
Enter your choice :
4
Select the column to be updated :
1. SALARY
2. BONUS
3. ROLE
Enter your choice :
1
SALARY UPDATE MENU
1. Increment salary for everyone
2. Update a new salary for an employee
Enter your choice :
1
Enter the increment amount :
1000
2 rows updated.
Would you like to continue?(y/n) :
y
Select the operation :
SQL MENU
1. Display records
2. Add record
3. Delete record
4. Update record
5. Count employees
6. Calculate pay
Enter your choice :
4
Select the column to be updated :
1. SALARY
2. BONUS
3. ROLE
Enter your choice :
2
Enter the new bonus for all employees :
13
2 rows updated.
Would you like to continue?(y/n) :
y
```



```
Select the operation :
SQL MENU
1. Display records
2. Add record
3. Delete record
4. Update record
5. Count employees
6. Calculate pay
Enter your choice :
4
Select the column to be updated :
1. SALARY
2. BONUS
3. ROLE
Enter your choice :
3
Enter the EID of the employee :
102
Enter the new ROLE :
CEO
1 row updated
Would you like to continue?(y/n) :
y
Select the operation :
SQL MENU
1. Display records
2. Add record
3. Delete record
4. Update record
5. Count employees
6. Calculate pay
Enter your choice :
1
Row 1 :
EID    : 101
ENAME  : Sachin
SALARY : 150001000
BONUS  : 13
ROLE   : Leader

Row 2 :
EID    : 102
ENAME  : Dhoni
SALARY : 140001000
BONUS  : 13
ROLE   : CEO

Would you like to continue?(y/n) :
y
```



```
Select the operation :
SQL MENU
1. Display records
2. Add record
3. Delete record
4. Update record
5. Count employees
6. Calculate pay
Enter your choice :
3
Enter the EID of the record to be deleted :
102
1 row deleted.
Would you like to continue?(y/n) :
y
Select the operation :
SQL MENU
1. Display records
2. Add record
3. Delete record
4. Update record
5. Count employees
6. Calculate pay
Enter your choice :
1
Row 1 :
EID    : 101
ENAME  : Sachin
SALARY : 150001000
BONUS  : 13
ROLE   : Leader

Would you like to continue?(y/n) :
n
```

RESULT:

The JAVA application for employee information system was implemented and executed successfully.

Exp No.: 10

Develop Java Networking programs

1. Write a TCP client/server system to illustrate two-way communication.

AIM:

To write a TCP client/server system to illustrate two-way communication.

ALGORITHM:

a) Server:

Step 1: Start.
Step 2: Import required packages.
Step 3: Create a server socket to accept client's request.
Step 4: Get the name and age of the user from the client.
Step 5: If age>=18, send a string granting access to the client.
Step 6: Else, send a string denying access to the client.
Step 7: Close the server.
Step 8: End.

b) Client:

Step 1: Start.
Step 2: Import required packages.
Step 3: Create a socket for the client script.
Step 4: Get the name and age of the user and send it to the server.
Step 5: Fetch and print the string sent by the server regarding the access.
Step 6: Close the client.
Step 7: End.

PROGRAM:

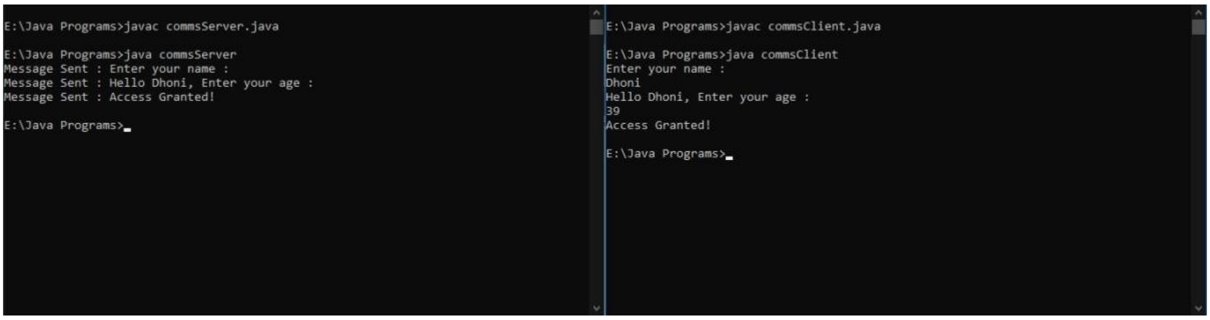
1 Server:

```
import  java.io.*;
import java.net.*;
public class commsServer
{
    public static void main(String[] args)
    {
        try
        {
            ServerSocket ss=new ServerSocket(3002);
            Socket s=ss.accept();
            DataOutputStream dout=new DataOutputStream(s.getOutputStream());
            DataInputStream din=new DataInputStream(s.getInputStream());
            String msg1="Enter your name : ";
            dout.writeUTF(msg1);
            System.out.println("Message Sent : "+msg1);
            String name=din.readUTF();
            String msg2="Hello "+name+", Enter your age : ";
            dout.writeUTF(msg2);
            System.out.println("Message Sent : "+msg2);
            int age=din.readInt();
            String msg3;
            if(age>=18)
                msg3="Access Granted!";
            else
                msg3="Access Denied! Underage!";
            dout.writeUTF(msg3);
            System.out.println("Message Sent : "+msg3);
            dout.close();
            din.close();
            s.close();
            ss.close();
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

2 Client:

```
import java.io.*;
import java.net.*;
import java.util.Scanner;
public class commsClient
{
    public static void main(String[] args)
    {
        try
        {
            Socket s=new Socket("localhost",3002);
            Scanner sc=new Scanner(System.in);
            DataInputStream din=new DataInputStream(s.getInputStream());
            DataOutputStream dout=new DataOutputStream(s.getOutputStream());
            System.out.println(din.readUTF());
            String name=sc.nextLine();
            dout.writeUTF(name);
            System.out.println(din.readUTF());
            int age=sc.nextInt();
            dout.writeInt(age);
            System.out.println(din.readUTF());
            din.close();
            dout.close();
            s.close();
            sc.close();
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

OUTPUT:



```
E:\Java Programs>javac commsServer.java

E:\Java Programs>java commsServer
Message Sent : Enter your name :
Message Sent : Hello Dhoni, Enter your age :
Message Sent : Access Granted!

E:\Java Programs>_
```

```
E:\Java Programs>javac commsClient.java

E:\Java Programs>java commsClient
Enter your name :
Dhoni
Hello Dhoni, Enter your age :
39
Access Granted!

E:\Java Programs>_
```

RESULT:

The TCP client/server system to illustrate two-way communication was implemented and executed successfully.

2. Write a UDP client/server system to illustrate communication.

AIM:

To write a UDP client/server system to illustrate communication.

ALGORITHM:

a) Server:

Step 1: Start.
Step 2: Import required packages.
Step 3: Create a datagram socket for the server.
Step 4: Receive the datagram packet from the client and print the message in it.
Step 5: Continue the communication, until the client sends ‘quit’.
Step 6: Close the server.
Step 7: End.

b) Client:

Step 1: Start.
Step 2: Import required packages.
Step 3: Create a datagram socket for the client.
Step 4: Get the message from the user and send the datagram packet to the server.
Step 5: Continue the communication, until the user enters ‘quit’.
Step 6: Close the client.
Step 7: End.

PROGRAM:

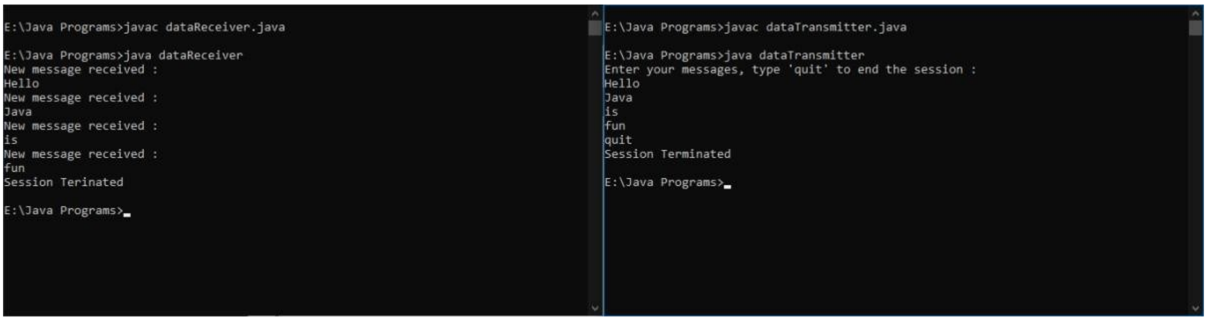
a) Server:

```
import java.net.*;
public class dataReceiver {
    public static void main(String[] args) {
        try
        {
            DatagramSocket ds=new DatagramSocket(3003);
            DatagramPacket dp;
            String msg;
            while(true)
            {
                byte[] receive= new byte[65535];
                dp=new DatagramPacket(receive, receive.length);
                ds.receive(dp);
                msg= new String(dp.getData(), 0, dp.getLength());
                if(msg.equals("quit"))
                {
                    System.out.println("Session Terinated");
                    break;
                }
                System.out.println("New message received : ");
                System.out.println(msg);
                receive=null;
            }
            ds.close();
        }
        catch (Exception e)
        {
            System.out.println(e);
        }
    }
}
```

b) Client:

```
import java.net.*;
import java.util.Scanner;
public class dataTransmitter {
    public static void main(String[] args) {
        try
        {
            Scanner s=new Scanner(System.in);
            DatagramSocket ds=new DatagramSocket();
            InetAddress ip=InetAddress.getLocalHost();
            byte buf[]=null;
            System.out.println("Enter your messages, type 'quit' to end the se
ssion : ");
            while(true)
            {
                String msg=s.nextLine();
                buf=msg.getBytes();
                DatagramPacket dp=new DatagramPacket(buf,buf.length,ip,3003);
                ds.send(dp);
                if(msg.equals("quit"))
                {
                    System.out.println("Session Terminated");
                    break;
                }
            }
            s.close();
            ds.close();
        }
        catch (Exception e)
        {
            System.out.println(e);
        }
    }
}
```

OUTPUT:



```
E:\Java Programs>javac dataReceiver.java

E:\Java Programs>java dataReceiver
New message received :
Hello
New message received :
Java
New message received :
is
New message received :
fun
Session Terinated

E:\Java Programs>
```

```
E:\Java Programs>javac dataTransmitter.java

E:\Java Programs>java dataTransmitter
Enter your messages, type 'quit' to end the session :
Hello
Java
is
fun
quit
Session Terminated

E:\Java Programs>
```

RESULT:

The UDP client/server system to illustrate communication was implemented and executed successfully.