Roll No.22239

Assignment 1: WAP to sort the elements of an array in ascending order.

Summary:

1) Array is nothing but the Collection of similar type of elements which has contiguous memory location.

2) Array is used to store a multiple values in a single variable.

3) To declare an Array, Define the variable type with square brackets

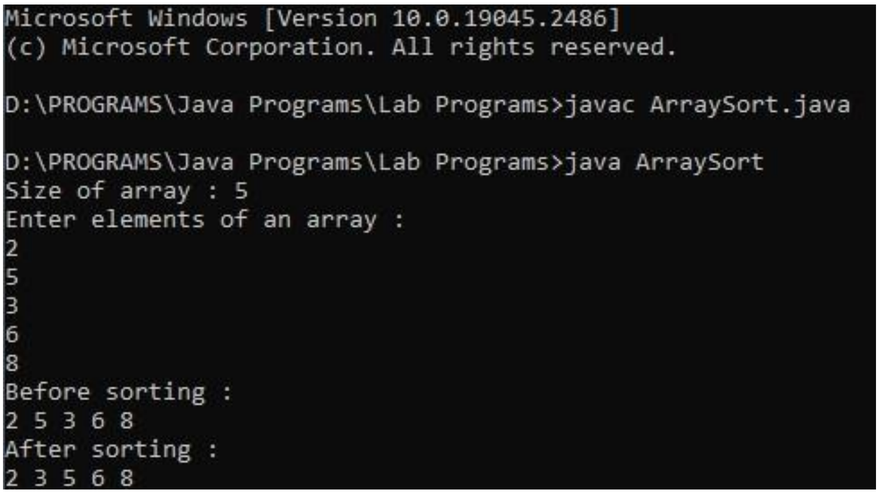
E.g.: For String: - String animals[ ]= new String[ ];

For Integer :- int array[ ] = new int[ ];

Source Code:

```
import java.util.Scanner;
class ArraySort{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Size of array : ");
        int n = sc.nextInt();
        int a[] = new int[n];           //create an array of size n
        System.out.println("Enter elements of an array : ");
        for(int i=0;i<n;i++){
            a[i] = sc.nextInt();        //Insert elements into the array
        }
        //Printing an original array before sorting
        System.out.println("Before sorting : ");
        for(int i=0;i<n;i++){
            System.out.print(a[i]+" ");
        }
        //Code to sort array elements in ascending order
        int temp = a[0];
        for(int i=0;i<n;i++){
            for(int j=i+1;j<n;j++){
                if(a[i]>a[j]){
                    temp = a[i];
                    a[i] = a[j];
                    a[j] = temp;
                }
            }
        }
        //Printing an array after sorting
        System.out.println("\nAfter sorting : ");
        for(int i=0;i<n;i++){
            System.out.print(a[i]+" ");
        }
    }
}
```

Output:



```
Microsoft Windows [Version 10.0.19045.2486]
(c) Microsoft Corporation. All rights reserved.

D:\PROGRAMS\Java Programs\Lab Programs>javac ArraySort.java

D:\PROGRAMS\Java Programs\Lab Programs>java ArraySort
Size of array : 5
Enter elements of an array :
2
5
3
6
8
Before sorting :
2 5 3 6 8
After sorting :
2 3 5 6 8
```

Roll No.22239

Assignment 2: WAP to find the transpose of a given matrix.

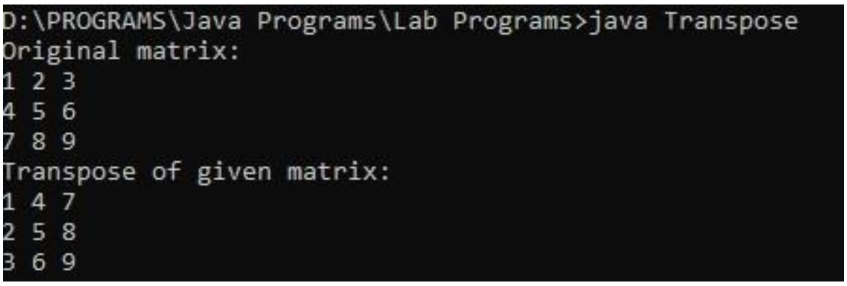
Summary:

Transpose of a Matrix is defined as “A Matrix which is formed by turning all the rows of a given matrix into columns and vice-versa.”

Source Code:

```
public class Transpose
{
   public static void main(String[] args) {
      int rows, cols;

      //Initialize matrix a
        int a[][] = {
                    {1, 2, 3},
                    {4, 5, 6},
                    {7, 8, 9}
                  };

        //Calculates number of rows and columns present in given matrix
      rows = a.length;
      cols = a[0].length;

      //Declare array t with reverse dimensions
      int t[][] = new int[cols][rows];

      //Calculates transpose of given matrix
      for(int i = 0; i < cols; i++){
         for(int j = 0; j < rows; j++){
            //Converts the row of original matrix into column of transposed matrix
            t[i][j] = a[j][i];
         }
      }
      // Printing Matrix without transpose:
      System.out.println("Original matrix: ");
      for(int i = 0; i < rows; i++){
         for(int j = 0; j < cols; j++){
           System.out.print(a[i][j] + " ");
         }
         System.out.println();
      }
      // Printing Matrix with transpose:
      System.out.println("Transpose of given matrix: ");
      for(int i = 0; i < cols; i++){
         for(int j = 0; j < rows; j++){
                System.out.print(t[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

Output:



```
D:\PROGRAMS\Java Programs\Lab Programs>java Transpose
Original matrix:
1 2 3
4 5 6
7 8 9
Transpose of given matrix:
1 4 7
2 5 8
3 6 9
```

Roll No.22239

Assignment 3: Wap to How to reverse a given String?
(Without using pre-defined function)
For example : Input = "abcd" Output = "dcba"

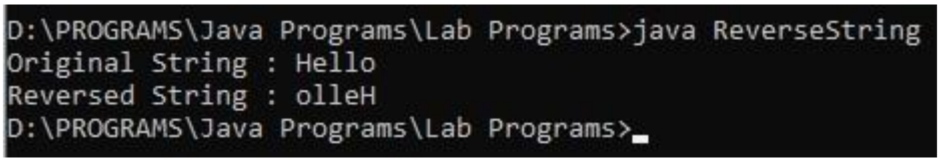
Summary:

We have several ways to reverse the string.

Source Code:

```
public class ReverseString {
    public static void main(String[] args) {
        String str = "Hello", r_str = "";
        //Printing Original String
        System.out.println("Original String : "+str);

        //Code to reverse a string
        for(int i=str.length()-1;i>=0;i--){
            r_str = r_str + str.charAt(i);
        }
        str = r_str;
        //Printing reversed string
        System.out.print("Reversed String : "+str);
    }
}
```

Output:



```
D:\PROGRAMS\Java Programs\Lab Programs>java ReverseString
Original String : Hello
Reversed String : olleH
D:\PROGRAMS\Java Programs\Lab Programs>_
```

Roll No.22239

Assignment 4: WAP to How do you check if a given String is Palindrome or not?

For example: "121" is a palindrome, but "123" is not. Your function should return true if given String is a palindrome, false otherwise Palindrome Strings are ="madam”, "anna", "nitin".

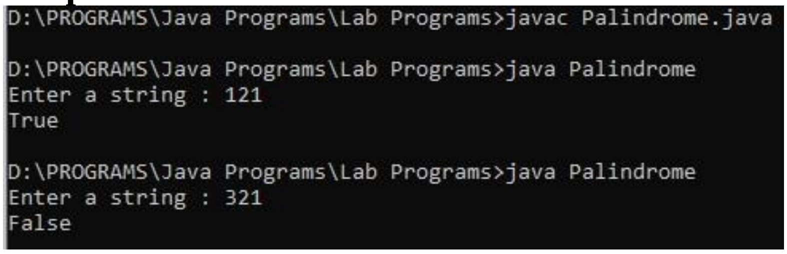
Summary:

A palindrome number is a number that is same after reverse. For example 545, 151, 34543, 343, 171, 48984 are the palindrome numbers. It can also be a string like LOL, MADAM etc.

Source Code:

```
import java.util.Scanner;
public class Palindrome {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string : ");
        String str = sc.next();
        str = str.toLowerCase();
        int flag = 0, i=0, len = str.length()-1;
        while(len > i){
            //compare a character present at starting and ending of the string
            if(str.charAt(i) != str.charAt(len)){
                flag = 1;   //Not a Palindrome
                break;
            }
            len--;//decrease ending length
            i++;//increase starting length
        }
        if(flag==0){
            System.out.println("True");     //Palindrome
        }
        else{
        System.out.println("False");        //Not a Palindrome
        }
    }
}
```

Output



```
D:\PROGRAMS\Java Programs\Lab Programs>javac Palindrome.java

D:\PROGRAMS\Java Programs\Lab Programs>java Palindrome
Enter a string : 121
True

D:\PROGRAMS\Java Programs\Lab Programs>java Palindrome
Enter a string : 321
False
```

Roll No.22239

Assignment 5: Create a package mca1 which will have 2 classes as class Mathematics with a methods to add two numbers, add three float numbers and class Maximum with a method to find maximum of three numbers.

Summary:

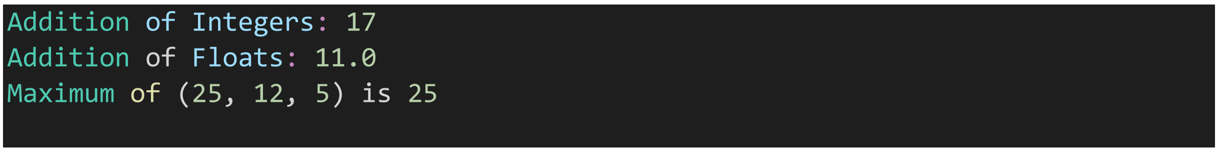
A package in Java is used to group related classes. Think of it as a folder in a file directory. We use packages to avoid name conflicts and to write better maintainable code.

Source Code:

```
package mca1;
public class Mathematics {
   //Method to add 2 numbers
   public int add(int a, int b) {
      return a+b;
   }
   //Method to add 3 float numbers
   public float add(float a, float b, float c) {
      return a+b+c;
   }
}

package mca1;
public class Maximum {
   //Method to find maximum among three
   public void max(int a, int b, int c){
      if(a>b && a>c)
         System.out.println("Maximum of ("+a+", "+b+", "+c+") is "+a);
      else if(b>a && b>c)
         System.out.println("Maximum of ("+a+", "+b+", "+c+") is "+b);
      else
         System.out.println("Maximum of ("+a+", "+b+", "+c+") is "+c);
   }
}

import mca1.*;
class Package{
   public static void main(String[] args) {
      //Mathematics class
      Mathematics m1 = new Mathematics();
      int add1 = m1.add(12,5);        //Addition of two numbers
      System.out.println("Addition of Integers: "+add1);

      Mathematics m2 = new Mathematics();
      float p = 2.5f, q = 3.5f, r = 5f;
      float add2 = m2.add(p, q, r);   // Addition three float numbers
      System.out.println("Addition of Floats: "+add2);
```

```
        //Maximum Class
        Maximum m = new Maximum();
        m.max(25,12, 5);                //Find maximum of three numbers
        // System.out.println("Addition : "+a);
    }
}
```

Output:



```
Addition of Integers: 17
Addition of Floats: 11.0
Maximum of (25, 12, 5) is 25
```

Roll No.22239

Assignment 6: Write a Java program to create an Animal interface that contains run() and eat() method. And implement methods in Dog and Cat class.

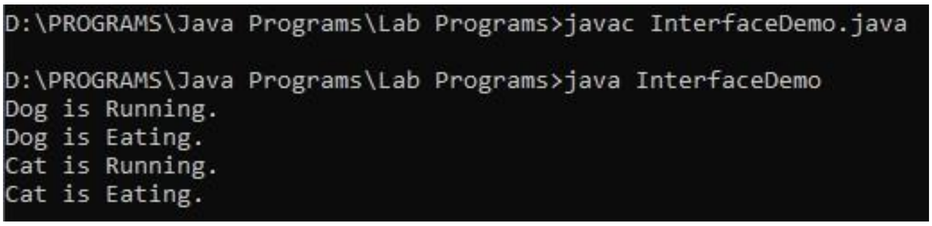
Summary:

An Interface in Java programming is defined as an abstract type used to specify the behavior of a class. A Java interface contains static constants and abstract methods. A class can implement multiple interfaces. In Java, interfaces are declared using the interface keyword.

Source Code:

```
// An interface is a completely "abstract class"
// Interfaces cannot be used to create objects.Therefore it cannot contain a constructor
// Interface methods do not have a body
// Interface methods are by default abstract and public
interface Animal{           // Animal Interface
   // Method run & eat are by default public
   void run();
   void eat();
}
//Class Dog which implements Animal interface
class Dog implements Animal{
   //On implementation of an interface, we must override all of its methods
   public void run(){
      System.out.println("Dog is Running.");
   }

   public void eat(){
      System.out.println("Dog is Eating.");
   }
}
//Class Cat which implements Animal interface
class Cat implements Animal{
   //On implementation of an interface, we must override all of its methods
   public void run(){
      System.out.println("Cat is Rinning.");
   }

   public void eat(){
      System.out.println("Cat is Eating.");
   }
}
//Class Interface Demo
class InterfaceDemo{
   public static void main(String[] args) {
      Dog d = new Dog();
      d.run();
      d.eat();

      Cat c = new Cat();
      c.run();
      c.eat();
   }
}
```

Output:



```
D:\PROGRAMS\Java Programs\Lab Programs>javac InterfaceDemo.java

D:\PROGRAMS\Java Programs\Lab Programs>java InterfaceDemo
Dog is Running.
Dog is Eating.
Cat is Running.
Cat is Eating.
```

Roll No.22239

Assignment 7: Write a Java program to create an abstract class Animal that contains non abstract run() and abstract eat() method…Derive two classes Dog and Cat from it.

Summary:

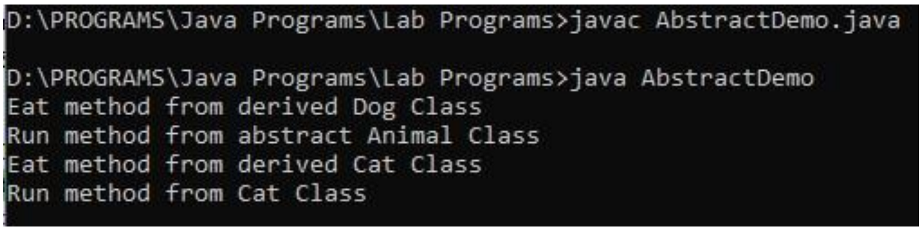
An abstract class allows you to create functionality that subclasses can implement or override. An interface only allows you to define functionality, not implement it. And whereas a class can extend only one abstract class, it can take advantage of multiple interfaces

Source Code:

```
// Data abstraction is the process of hiding certain details and showing only essential
information to the user.
// Abstraction can be achieved with either abstract classes or interfaces
// Abstract class: is a restricted class that cannot be used to create objects (to access it, it must
be inherited from another class).
// Abstract method: can only be used in an abstract class, and it does not have a body. The
body is provided by the subclass (inherited from).

abstract class Animal{      // Abstract class
   void run(){              // Non-abstract method
      System.out.println("Run method from abstract Animal Class");
   }

   abstract void eat();    // Abstract method which does not have a method defination
}
class Dog extends Animal{   // Dog class which is derived from Animal class
   void eat(){             // Mandatory to write defination for abstract method
      System.out.println("Eat method from derived Dog Class");
   }
}
class Cat extends Animal{   // Cat class which is derived from Animal class
   void eat(){             // Mandatory to write defination for abstract method
      System.out.println("Eat method from derived Cat Class");
   }
   void run(){
      System.out.println("Run method from Cat Class");
   }
}
// Main class
class AbstractDemo{
   public static void main(String[] args) {
      Dog d = new Dog();
      d.eat();
      d.run();
```

```
        Cat c = new Cat();
        c.eat();
        c.run();
    }
}
```

Output:



```
D:\PROGRAMS\Java Programs\Lab Programs>javac AbstractDemo.java

D:\PROGRAMS\Java Programs\Lab Programs>java AbstractDemo
Eat method from derived Dog Class
Run method from abstract Animal Class
Eat method from derived Cat Class
Run method from Cat Class
```

Roll No.22239

Assignment 8: Write a Java program to test any five of standard exception.

Summary:

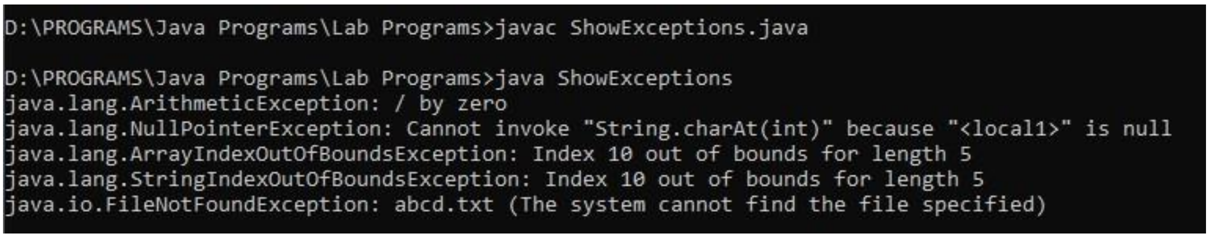
An exception is an unwanted or unexpected event, which occurs during the execution of a program i.e. at run time, which disrupts the normal flow of the program's instructions.

An exception handling is managed via five keywords: try, catch, throw, throws, and finally.

Source Code:

```
import java.io.*;
public class ShowExceptions {
    public static void main(String[] args) {
        //1. ArithmeticException
        try{
            int a = 5, b =0, sum = 0;
            sum = a/b;                      //sum = 5/0 which is not possible
            System.out.println("Adition : "+sum);
        }
        catch(ArithmeticException e){
            System.out.println(e);
        }
        //2. NullPointerException
        try{
            String str = null;              //string is null
            System.out.println("String : "+str.charAt(0));
        }
        catch(NullPointerException e){
            System.out.println(e);
        }
        //3. ArrayIndexOutOfBoundsException
        try{
            int arr[] = {0,1,2,3,4};        //length of the array is 5
            System.out.println(arr[10]);
        }
        catch(ArrayIndexOutOfBoundsException e){
            System.out.println(e);
        }
        //4. StringIndexOutOfBoundsException
        try{
            String str = "Sahil";           //string length is 7
            System.out.println(str.charAt(10));
        }
        catch(StringIndexOutOfBoundsException e){
            System.out.println(e);
        }
        //5. FileNotFoundException
        try{
            File myFile = new File("abcd.txt"); //file abcd.txt does not exist
            FileReader myReader = new FileReader(myFile);
        }
        catch(FileNotFoundException fe){
            System.out.println(fe);
        }
    }
}
```

Output:



```
D:\PROGRAMS\Java Programs\Lab Programs>javac ShowExceptions.java

D:\PROGRAMS\Java Programs\Lab Programs>java ShowExceptions
java.lang.ArithmeticException: / by zero
java.lang.NullPointerException: Cannot invoke "String.charAt(int)" because "<local1>" is null
java.lang.ArrayIndexOutOfBoundsException: Index 10 out of bounds for length 5
java.lang.StringIndexOutOfBoundsException: Index 10 out of bounds for length 5
java.io.FileNotFoundException: abcd.txt (The system cannot find the file specified)
```

Roll No.22239

Assignment 9: Write a class Student with attributes roll no, name, age and course. Accept the details of Student from the user. If age of student is not in between 18 and 28 then throw user-defined exception “Age Not Within The Range”. Otherwise display the information of students.

Summary:

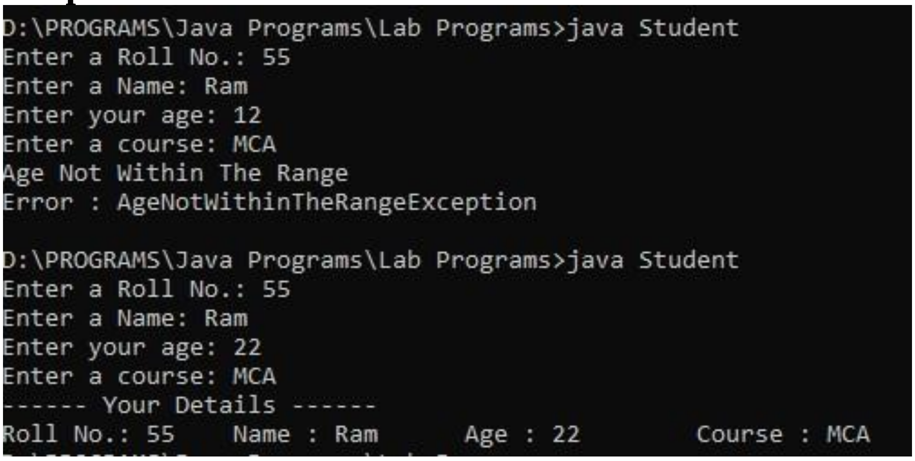
User Defined Exception or custom exception is creating your own exception class and throws that exception using **'throw'** keyword. This can be done by extending the class **Exception**.

Source Code:

```
import java.util.Scanner;
class Student{
    public static void main(String[] args) {
        String name, course;
        int rollno, age;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a Roll No.: ");
        rollno = sc.nextInt();

        System.out.print("Enter a Name: ");
        name = sc.next();

        System.out.print("Enter your age: ");
        age = sc.nextInt();

        System.out.print("Enter a course: ");
        course = sc.next();
        try{
            if(age<18 || age>28)
                throw new AgeNotWithinTheRangeException();
            else{
                System.out.println("------ Your Details ------");
                System.out.print("Roll No.: "+rollno);
                System.out.print("\tName : "+name);
                System.out.print("\tAge : "+age);
                System.out.print("\tCourse : "+course);
            }

        }
        catch(Exception e){
            System.out.println("Error : "+e);
        }
    }
}
// User Defined Exception
```

```
class AgeNotWithinTheRangeException extends Exception{
    AgeNotWithinTheRangeException(){
        System.out.println("Age Not Within The Range");
    }
}
```

Output:



```
D:\PROGRAMS\Java Programs\Lab Programs>java Student
Enter a Roll No.: 55
Enter a Name: Ram
Enter your age: 12
Enter a course: MCA
Age Not Within The Range
Error : AgeNotWithinTheRangeException

D:\PROGRAMS\Java Programs\Lab Programs>java Student
Enter a Roll No.: 55
Enter a Name: Ram
Enter your age: 22
Enter a course: MCA
------ Your Details ------
Roll No.: 55    Name : Ram      Age : 22        Course : MCA
```

Roll No.22239

Assignment 10: Write a Java program to create a Thread by extending the Thread class. And print the name of currently executing thread.

Summary:

Threads allows a program to operate more efficiently by doing multiple things at the same time. Threads can be used to perform complicated tasks in the background without interrupting the main program.

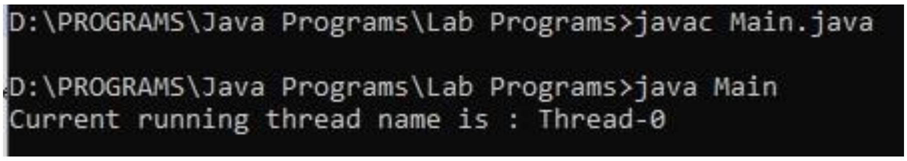
There are two ways to create a thread.

1. It can be created by extending the Thread class and overriding its run() method

Source Code:

```
public class Main extends Thread{
    public void run(){
        System.out.println("Current running thread name is : 
"+Thread.currentThread().getName());
    }
    public static void main(String[] args) {
        Main m = new Main();
        m.start();
    }
}
```

Output:



```
D:\PROGRAMS\Java Programs\Lab Programs>javac Main.java

D:\PROGRAMS\Java Programs\Lab Programs>java Main
Current running thread name is : Thread-0
```

Roll No.22239

Assignment 11: Write a Java program to create a Thread by Implementing the Runnable Interface. And print the name of currently executing thread.

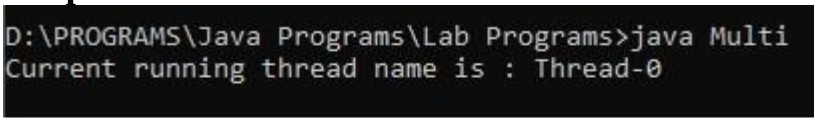
Summary:

1. Another way to create a thread is to implement the Runnable interface

Source Code:

```
public class Multi implements Runnable {
    public void run(){
        System.out.println("Current running thread name is : 
"+Thread.currentThread().getName());
    }
    public static void main(String[] args) {
        Multi m = new Multi();
        Thread t = new Thread(m);
        t.start();
    }
}
```

Output:



```
D:\PROGRAMS\Java Programs\Lab Programs>java Multi
Current running thread name is : Thread-0
```

Roll No.22239

Assignment 12: Write a multithreaded program to print even and odd numbers. Create two threads, one thread prints even number and second thread prints odd number.

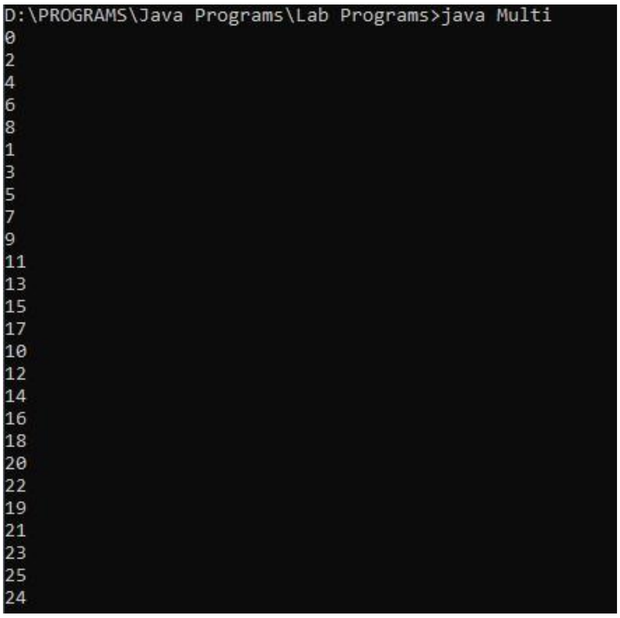
Summary:

Multithread program to print even and odd numbers .create two thread, one thread print even number and second thread print odd number.

Source Code:

```
public class Multi {

   public static void main(String[] args) {
      Thread1 t1 = new Thread1();
      Thread2 t2 = new Thread2();
      t1.start();
      t2.start();
   }
}


class Thread1 extends Thread {
   public void run() {

      for (int i = 0; i <= 25; i++) {
         if (i % 2 == 0)
            System.out.println(i);
      }
   }
}

class Thread2 extends Thread {
   public void run() {

      for (int i = 0; i <= 25; i++) {
         if (i % 2 != 0)
            System.out.println(i);
      }
   }
}
```

Output:



```
D:\PROGRAMS\Java Programs\Lab Programs>java Multi
0
2
4
6
8
1
3
5
7
9
11
13
15
17
10
12
14
16
18
20
22
19
21
23
25
24
```

Roll No.22239

Assignment 13: Write a code to remove duplicates from Array List in Java.

Summary:

The ArrayList class is a resizable array, which can be found in the java.util package.

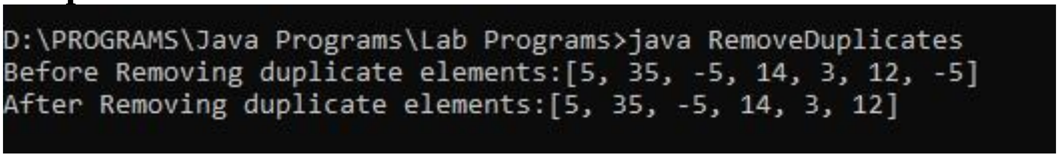
The difference between a built-in array and an ArrayList in Java, is that the size of an array cannot be modified (if you want to add or remove elements to/from an array, you have to create a new one). While elements can be added and removed from an ArrayList whenever you want.

The ArrayList class has a regular array inside it. When an element is added, it is placed into the array. If the array is not big enough, a new, larger array is created to replace the old one and the old one is removed.

Source Code:

```
import java.util.ArrayList;

class RemoveDuplicates{
   public static void main(String[] args) {
      ArrayList<Integer> arr = new ArrayList<Integer>();

      //arr.add(value) - To add value into the ArrayList
      arr.add(5);
      arr.add(35);
      arr.add(-5);
      arr.add(14);
      arr.add(3);
      arr.add(12);
      arr.add(-5);

      //Printing original arraylist items
      System.out.println("Before Removing duplicate elements:"+arr);

      //code to remove duplicate elements from an ArrayList arr
      for(int i=0;i<arr.size();i++){      //arr.size() - To get size of an ArrayList
         for(int j=i+1;j<arr.size();j++){
         if(arr.get(i).equals(arr.get(j))){
                  //arr.get(index) - To get an ArrayList element present at sepecified index
               arr.remove(j); //arr.remove(index) - Remove element present at specified index
            }
         }
      }
      System.out.println("After Removing duplicate elements:"+arr);
   }
}
```

Output:



```
D:\PROGRAMS\Java Programs\Lab Programs>java RemoveDuplicates
Before Removing duplicate elements:[5, 35, -5, 14, 3, 12, -5]
After Removing duplicate elements:[5, 35, -5, 14, 3, 12]
```

Roll No.22239

Assignment 14: Write a code to sort a linked list and Reverse a linked list in java.

Summary:

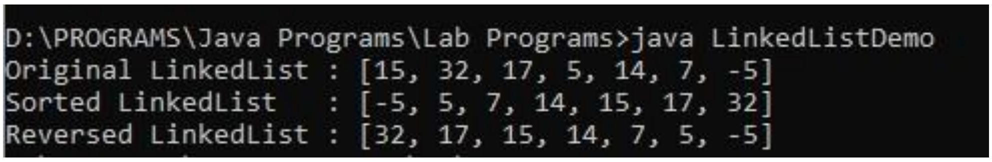
The LinkedList class is almost identical to the ArrayList

The LinkedList stores its items in "containers." The list has a link to the first container and each container has a link to the next container in the list. To add an element to the list, the element is placed into a new container and that container is linked to one of the other containers in the list.

Source Code:

```
import java.util.LinkedList;
public class LinkedListDemo {
    public static void main(String[] args) {
        LinkedList<Integer> list = new LinkedList<Integer>();
        LinkedList<Integer> ReversedList = new LinkedList<Integer>();

        //list.add(element) - To add element into the LinkedList
        list.add(15);
        list.add(32);
        list.add(17);
        list.add(5);
        list.add(14);
        list.add(7);
        list.add(-5);

        //Printing original LinkedList elements
        System.out.print("Original LinkedList : "+list);

        //code to sort an elements of LinkedList
        for(int i=0;i<list.size();i++){                    //list.size() - To get size of an LinkedList
            for(int j=i+1;j<list.size();j++){
                if(list.get(i) > list.get(j)){
                                //list.get(index) - To get an LinkedList element present at sepecified
                        index
                    int temp = list.get(i);
                    list.set(i, list.get(j));
                    //list.set(index, element) - To replace the element at the specified position in this
list with the specified element.
                    list.set(j, temp);
                }
            }
        }
        //Printing Sorted LinkedList elements
        System.out.print("\nSorted LinkedList   : "+list);

        //Code to Reverse a LinkedList
        for(int i=list.size()-1;i>=0;i--){
           ReversedList.add(list.get(i));
        }

        //Printing Reversed LinkedList elements
        System.out.print("\nReversed LinkedList : "+ReversedList);

    }
}
```

Output:



```
D:\PROGRAMS\Java Programs\Lab Programs>java LinkedListDemo
Original LinkedList : [15, 32, 17, 5, 14, 7, -5]
Sorted LinkedList   : [-5, 5, 7, 14, 15, 17, 32]
Reversed LinkedList : [32, 17, 15, 14, 7, 5, -5]
```

Roll No.22239

Assignment 15: Write a Java program to copy the contents of a file to another file.

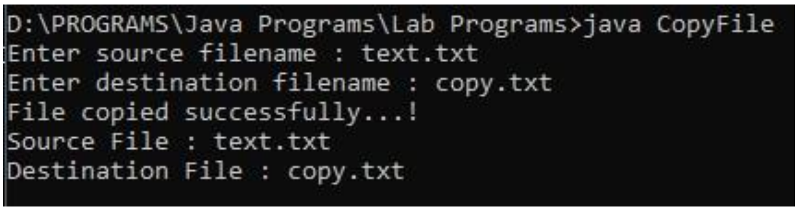
Summary:

Java File class represents the files and directory pathnames in an abstract manner. This class is used for creation of files and directories, file searching, file deletion, etc. The File object represents the actual file/directory on the disk.
The File class from the java.io package, allows us to work with files.

Source Code:

```
import java.io.*;           // Import the File class
import java.util.Scanner;  // Import the Scanner class to read text files
public class CopyFile {
    public static void main(String[] args) {

        //Get source filename and destination file name
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter source filename : ");
        String s_file = sc.next();
        System.out.print("Enter destination filename : ");
        String d_file = sc.next();

        //FileReader and FileWriter throws an exception.
        //Therefore code should be in a try-catch block
        try{
            File myFile = new File(s_file);
            //we use the Scanner class to read the contents of the text file
            Scanner myReader = new Scanner(myFile);
            String data = "";
            //myReader.hasNextLine() - To check file hasNextLine or not
            while(myReader.hasNextLine()){
                data = data + myReader.nextLine() + "\n";
            }

   //FileWriter class is used to write character-oriented data to a file. It also creates a new file.
            FileWriter myWriter = new FileWriter(d_file);
            myWriter.write(data);

            System.out.print("File copied successfully...!");
            System.out.print("\nSource File : "+s_file);
            System.out.println("\nDestination File : "+d_file);

            myWriter.close();                              //Close the FileWriter.
            myReader.close();                               //Close the FileReader.
        }
        catch(Exception e){
            System.out.println(e);
        }
    }
}
```

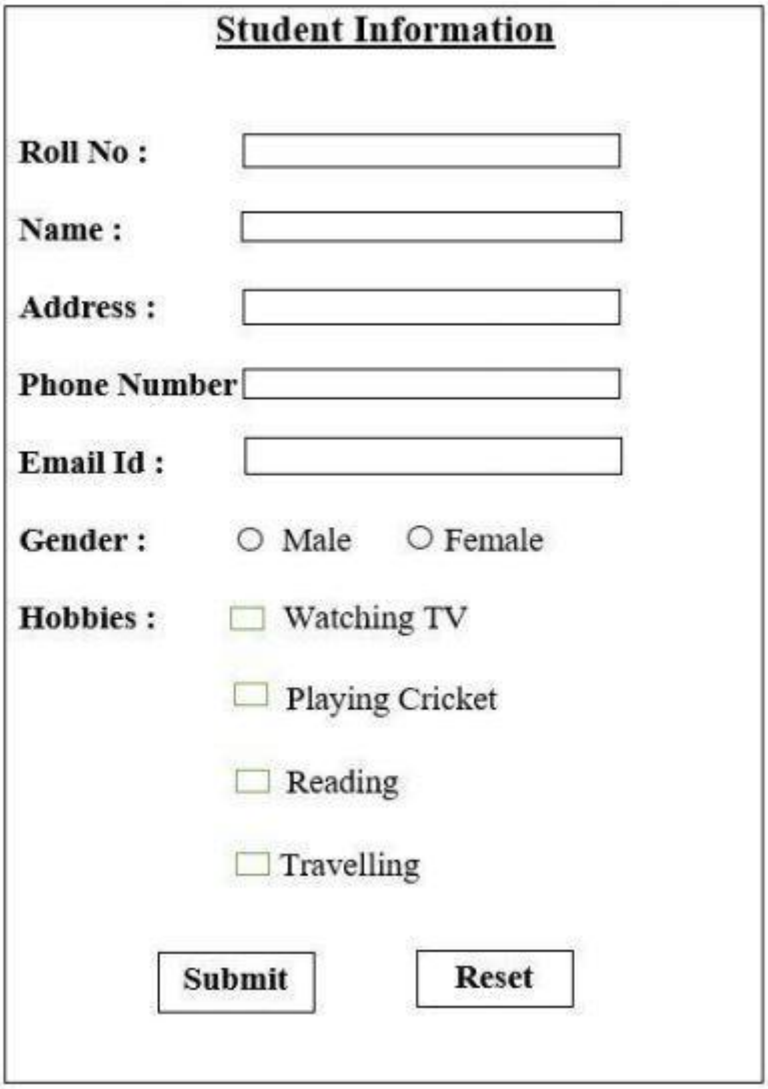
Output:



```
D:\PROGRAMS\Java Programs\Lab Programs>java CopyFile
Enter source filename : text.txt
Enter destination filename : copy.txt
File copied successfully...!
Source File : text.txt
Destination File : copy.txt
```

Roll No.22239

Assignment 16: Write Java AWT code to accept Student information and display Student details on the Screen.



Student Information

Roll No :

Name :

Address :

Phone Number

Email Id :

Gender : ○ Male ○ Female

Hobbies : □ Watching TV

□ Playing Cricket

□ Reading

□ Travelling

Summary:

AWT stands for Abstract window toolkit is an Application programming interface (API) for creating Graphical User Interface (GUI) in Java. It allows Java programmers to develop window-based applications. AWT provides various components like button, label, checkbox, etc. used as objects inside a Java Program.

Abstract Window Toolkit
The Abstract Window Toolkit (AWT) supports Graphical User Interface (GUI) programming. AWT features include: A set of native user interface components.

Source Code:

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

class StudentInformation extends JFrame implements ActionListener {
    JLabel title, ul, l1, l2, l3, l4, l5, l6, l7, l8;
    JTextField t1, t2, t3, t4, t5, t6;
    JRadioButton jr1, jr2;
    JCheckBox ch1, ch2, ch3, ch4;
    JButton b1, b2;

    StudentInformation() {
        setLocation(550, 100); // Set Loaction of the Frame
        setSize(450, 580); // Set Size of the Frame

        // Create a new panel on the Frame
        JPanel p = new JPanel();
        p.setLayout(null);

        // create a label to display text
        title = new JLabel("Student Information");
        title.setBounds(130, 10, 300, 20); // Set label at (x axis, y axis, width, height)
        title.setFont(new Font("TimesRoman", Font.BOLD, 25)); // Set a font, thickness and
size
        p.add(title); // Add a Jlabel to the JPanel

        ul = new JLabel("________________________________");
        ul.setBounds(130, 18, 300, 20);
        ul.setFont(new Font("TimesRoman", Font.BOLD, 15));
        p.add(ul);

        l1 = new JLabel("Roll No :");
        l1.setBounds(20, 60, 100, 20);
        l1.setFont(new Font("TimesRoman", Font.BOLD, 20));
        p.add(l1);
        // create a TextField
        t1 = new JTextField();
        t1.setBounds(150, 60, 200, 20);
        p.add(t1);

        l2 = new JLabel("Name :");
        l2.setBounds(20, 100, 100, 20);
        l2.setFont(new Font("TimesRoman", Font.BOLD, 20));
        p.add(l2);

        t2 = new JTextField();
        t2.setBounds(150, 100, 200, 20);
        p.add(t2);
```

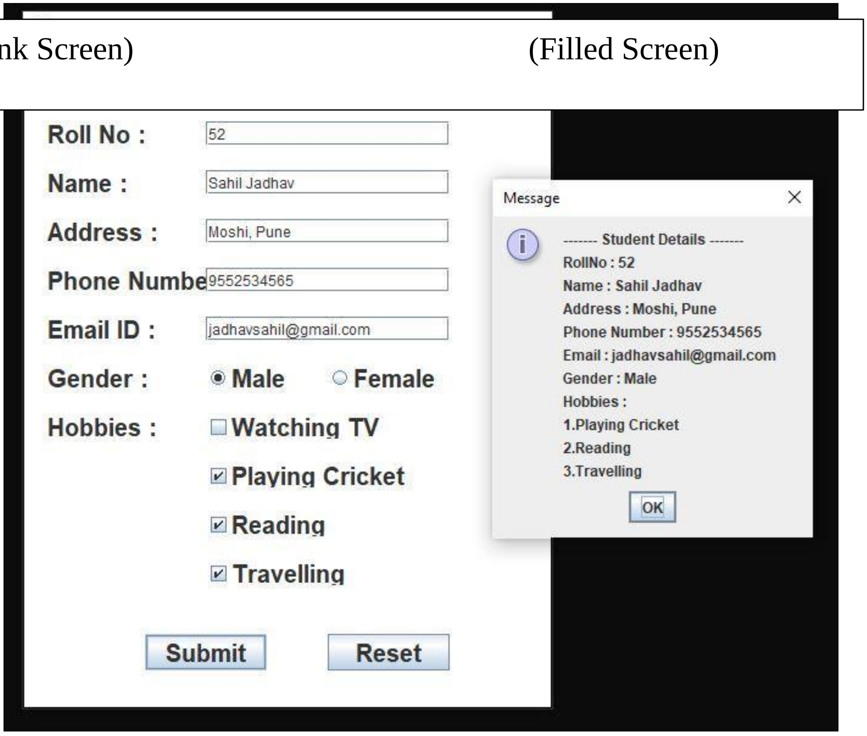
```
l3 = new JLabel("Address :");
l3.setBounds(20, 140, 100, 20);
l3.setFont(new Font("TimesRoman", Font.BOLD, 20));
p.add(l3);

t3 = new JTextField();
t3.setBounds(150, 140, 200, 20);
p.add(t3);

l4 = new JLabel("Phone Number :");
l4.setBounds(20, 180, 150, 20);
l4.setFont(new Font("TimesRoman", Font.BOLD, 20));
p.add(l4);

t4 = new JTextField();
t4.setBounds(150, 180, 200, 20);
p.add(t4);

l5 = new JLabel("Email ID :");
l5.setBounds(20, 220, 100, 20);
l5.setFont(new Font("TimesRoman", Font.BOLD, 20));
p.add(l5);

t5 = new JTextField();
t5.setBounds(150, 220, 200, 20);
p.add(t5);

l6 = new JLabel("Gender :");
l6.setBounds(20, 260, 100, 20);
l6.setFont(new Font("TimesRoman", Font.BOLD, 20));
p.add(l6);

// create a radio button
jr1 = new JRadioButton("Male");
jr1.setBounds(150, 260, 100, 20);
jr1.setFont(new Font("TimesRoman", Font.BOLD, 20));
jr1.setBackground(Color.WHITE);
p.add(jr1);

jr2 = new JRadioButton("Female");
jr2.setBounds(250, 260, 100, 20);
jr2.setFont(new Font("TimesRoman", Font.BOLD, 20));
jr2.setBackground(Color.WHITE);
p.add(jr2);

l8 = new JLabel("Hobbies :");
l8.setBounds(20, 300, 100, 20);
l8.setFont(new Font("TimesRoman", Font.BOLD, 20));
p.add(l8);
```

```
    // create a checkbox
    ch1 = new JCheckBox("Watching TV");
    ch2 = new JCheckBox("Playing Cricket");
    ch3 = new JCheckBox("Reading");
    ch4 = new JCheckBox("Travelling");
    ch1.setBounds(150, 300, 150, 20);
    ch2.setBounds(150, 340, 250, 20);
    ch3.setBounds(150, 380, 150, 20);
    ch4.setBounds(150, 420, 200, 20);
    ch1.setFont(new Font("TimesRoman", Font.BOLD, 20)); // setFont
    ch2.setFont(new Font("TimesRoman", Font.BOLD, 20)); // setFont
    ch3.setFont(new Font("TimesRoman", Font.BOLD, 20)); // setFont
    ch4.setFont(new Font("TimesRoman", Font.BOLD, 20)); // setFont
    ch1.setBackground(Color.WHITE); // Changing background color
    ch2.setBackground(Color.WHITE); // Changing background color
    ch3.setBackground(Color.WHITE); // Changing background color
    ch4.setBackground(Color.WHITE); // Changing background color
    p.add(ch1); // Add checkbox to the JPanel
    p.add(ch2); // Add checkbox to the JPanel
    p.add(ch3); // Add checkbox to the JPanel
    p.add(ch4); // Add checkbox to the JPanel

    // create a Submit Button
    b1 = new JButton("Submit");
    b1.setBounds(100, 480, 100, 30);
    b1.setFont(new Font("TimesRoman", Font.BOLD, 20));
    p.add(b1);

    // create a Reset Button
    b2 = new JButton("Reset");
    b2.setBounds(250, 480, 100, 30);
    b2.setFont(new Font("TimesRoman", Font.BOLD, 20));
    p.add(b2);

    // Set a Panel Background
    p.setBackground(Color.WHITE);
    p.setBorder(BorderFactory.createLineBorder(Color.black));// Set a border to the panel
    add(p); // Add JPanel on Frame
    b1.addActionListener(this);
    b2.addActionListener(this);

}

public void actionPerformed(ActionEvent e) {
    if (e.getSource() == b1){
        JOptionPane.showMessageDialog(null, "Submitted...!"); // Prompt a message screen
        String gender;
        String hobbies = "";
        int flag = 0;
        if(jr1.isSelected())
```

```
            gender = "Male";
        else if(jr2.isSelected())
            gender = "Female";
        else
            gender = "Not Selected";

        if(ch1.isSelected()){
            flag++;
            hobbies += "\n\t"+flag+"."+ch1.getText();
        }
        if(ch2.isSelected()){
            flag++;
            hobbies += "\n\t"+flag+"."+ch2.getText();
        }
        if(ch3.isSelected()){
            flag++;
            hobbies += "\n\t"+flag+"."+ch3.getText();
        }
        if(ch4.isSelected()){
            flag++;
            hobbies += "\n\t"+flag+"."+ch4.getText();
        }
        if(flag == 0)
            hobbies = "Not Selected";
        String details = "------- Student Details -------"+"\n"
                    +"RollNo : "+t1.getText()+"\n"
                    +"Name : "+t2.getText()+"\n"
                    +"Address : "+t3.getText()+"\n"
                    +"Phone Number : "+t4.getText()+"\n"
                    +"Email : "+t5.getText()+"\n"
                    +"Gender : "+gender+"\n"
                    +"Hobbies : "+hobbies+"\n";
        JOptionPane.showMessageDialog(null, details); // Prompt a message screen
    }
    else if (e.getSource() == b2)
        JOptionPane.showMessageDialog(null, "Cleared !!"); // Prompt a message screen
        t1.setText(null);
        t2.setText(null);
        t3.setText(null);
        t4.setText(null);
        t5.setText(null);
        jr1.setSelected(false);
        jr2.setSelected(false);
        ch1.setSelected(false);
        ch2.setSelected(false);
        ch3.setSelected(false);
        ch4.setSelected(false);
}

public static void main(String[] args) {
        new StudentInformation().setVisible(true);
    }
}
```

Output:

Blank Screen) (Filled Screen)



Roll No : 52

Name : Sahil Jadhav

Address : Moshi, Pune

Phone Numbe 9552534565

Email ID : jadhavsahil@gmail.com

Gender : ● Male ○ Female

Hobbies : □ Watching TV

☑ Playing Cricket

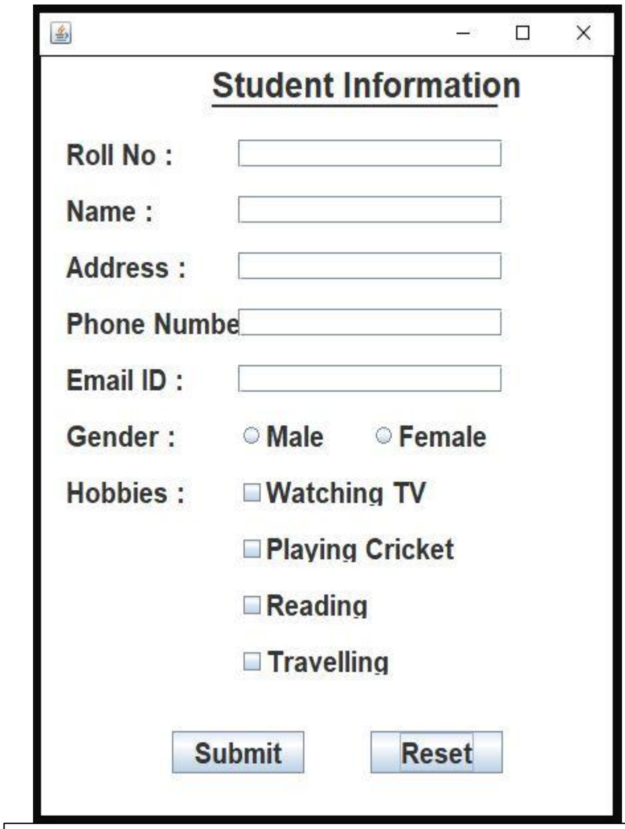
☑ Reading

☑ Travelling

Message

------- Student Details -------
RollNo : 52
Name : Sahil Jadhav
Address : Moshi, Pune
Phone Number : 9552534565
Email : jadhavsahil@gmail.com
Gender : Male
Hobbies :
1.Playing Cricket
2.Reading
3.Travelling

(Display Student Details)



Student Information

Roll No :

Name :

Address :

Phone Numbe

Email ID :

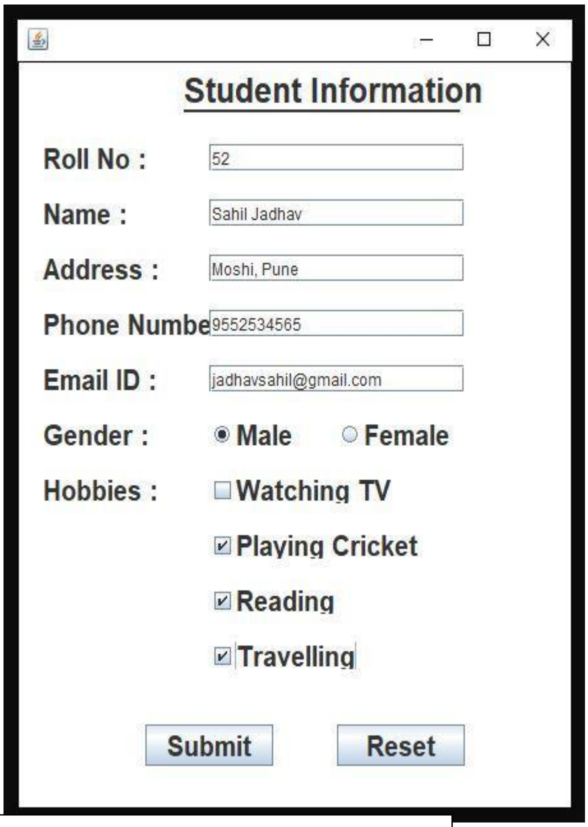
Gender : ○ Male ○ Female

Hobbies : □ Watching TV

□ Playing Cricket

□ Reading

□ Travelling



Student Information

Roll No : 52

Name : Sahil Jadhav

Address : Moshi, Pune

Phone Numbe 9552534565

Email ID : jadhavsahil@gmail.com

Gender : ● Male ○ Female

Hobbies : □ Watching TV

☑ Playing Cricket

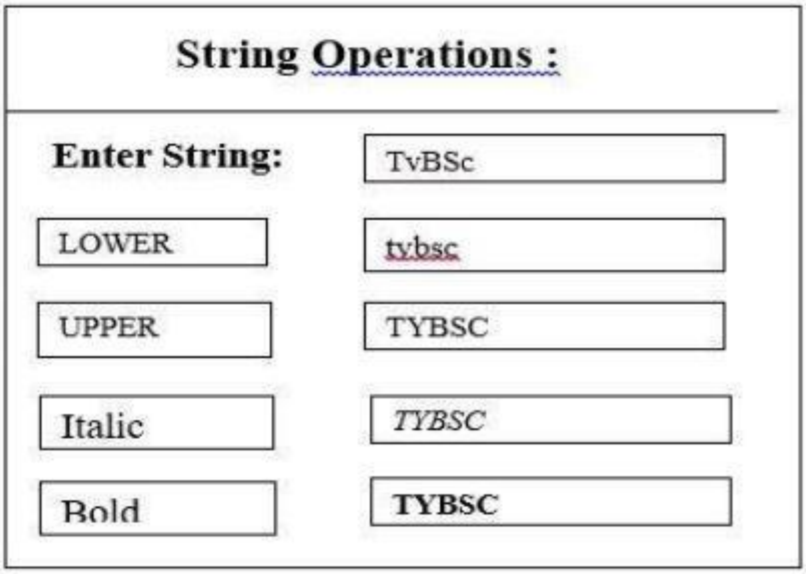
☑ Reading

☑ Travelling

(Submitted Screen) (Reset Screen)

Roll No.22239

Assignment 17: Write a JAVA program to design a screen using Swing to perform String operations.



Summary:

Swing in java is part of Java foundation class which is lightweight and platform independent. It is used for creating window based applications. It includes components like button, scroll bar, text field etc. Putting together all these components makes a graphical user interface.

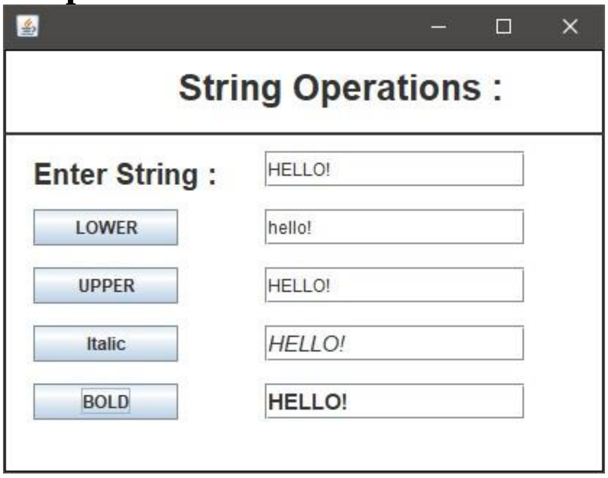
Swing in Java is a Graphical User Interface (GUI) toolkit that includes the GUI components. Swing provides a rich set of widgets and packages to make sophisticated GUI components for Java applications

Source Code:

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
public class StringOperations extends JFrame implements ActionListener {
    JLabel title, line, l1;
    JTextField t1, t2, t3, t4, t5;
    JButton b1, b2, b3, b4;

    StringOperations() {
        setLocation(250, 100); // Set Loaction of the Frame
        setSize(430, 330); // Set Size of the Frame

        // Create a new panel on the Frame
        JPanel p = new JPanel();
        p.setLayout(null);

        // create a label to display text
    title = new JLabel("String Operations :");
    title.setBounds(120, 10, 250, 30); // Set label at (x axis, y axis, width, height)
    title.setFont(new Font("TimesRoman", Font.BOLD, 25)); // Set a font, thickness and
size
    p.add(title); // Add a Jlabel to the JPanel

    line = new JLabel("______________________________________________________________________");
    line.setBounds(0, 40, 500, 20);
    line.setFont(new Font("TimesRoman", Font.BOLD, 15));
    p.add(line);

    l1 = new JLabel("Enter String :");
    l1.setBounds(20, 70, 150, 30);
    l1.setFont(new Font("TimesRoman", Font.BOLD, 20));
    p.add(l1);

    // create a TextField
    t1 = new JTextField();
    t1.setBounds(180, 70, 180, 25);
    p.add(t1);

    // create a Button
    b1 = new JButton("LOWER");
    b1.setBounds(20, 110, 100, 25);
    p.add(b1);

    t2 = new JTextField();
    t2.setBounds(180, 110, 180, 25);
    p.add(t2);

    b2 = new JButton("UPPER");
    b2.setBounds(20, 150, 100, 25);
    p.add(b2);

    t3 = new JTextField();
    t3.setBounds(180, 150, 180, 25);
    p.add(t3);

    b3 = new JButton("Italic");
    b3.setBounds(20, 190, 100, 25);
    p.add(b3);

    t4 = new JTextField();
    t4.setBounds(180, 190, 180, 25);
    p.add(t4);

    b4 = new JButton("BOLD");
    b4.setBounds(20, 230, 100, 25);
    p.add(b4);
```

```
        t5 = new JTextField();
        t5.setBounds(180, 230, 180, 25);
        p.add(t5);

        p.setBackground(Color.WHITE); // Set a Panel Background
        p.setBorder(BorderFactory.createLineBorder(Color.BLACK)); // Set a border to the
panel
        add(p); // Add JPanel on Frame

        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);
        b4.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e) {
        String str = t1.getText();
        if (e.getSource() == b1) {
            t2.setText(str.toLowerCase());  // Convert String to LowerCase
        } else if (e.getSource() == b2) {
            t3.setText(str.toUpperCase());  // Convert String to UpperCase
        } else if (e.getSource() == b3) {
            t4.setText(str.toUpperCase());  // Convert String to LowerCase
            t4.setFont(new Font("TimesRoman", Font.ITALIC, 15));    // ITALIC
        } else if (e.getSource() == b4) {
            t5.setText(str.toUpperCase());  // Convert String to LowerCase
            t5.setFont(new Font("TimesRoman", Font.BOLD, 15));  //BOLD
        }
    }

    public static void main(String[] args) {
        new StringOperations().setVisible(true);
    }
}
```

Output:



String Operations :

Enter String : HELLO!

hello!

HELLO!

HELLO!

HELLO!

Roll No.22239

Assignment 18: Write a Java code to Read, Insert, Update and Delete any record from the database. (Employee table contain details emp_no, emp_name, address, salary).

Summary:

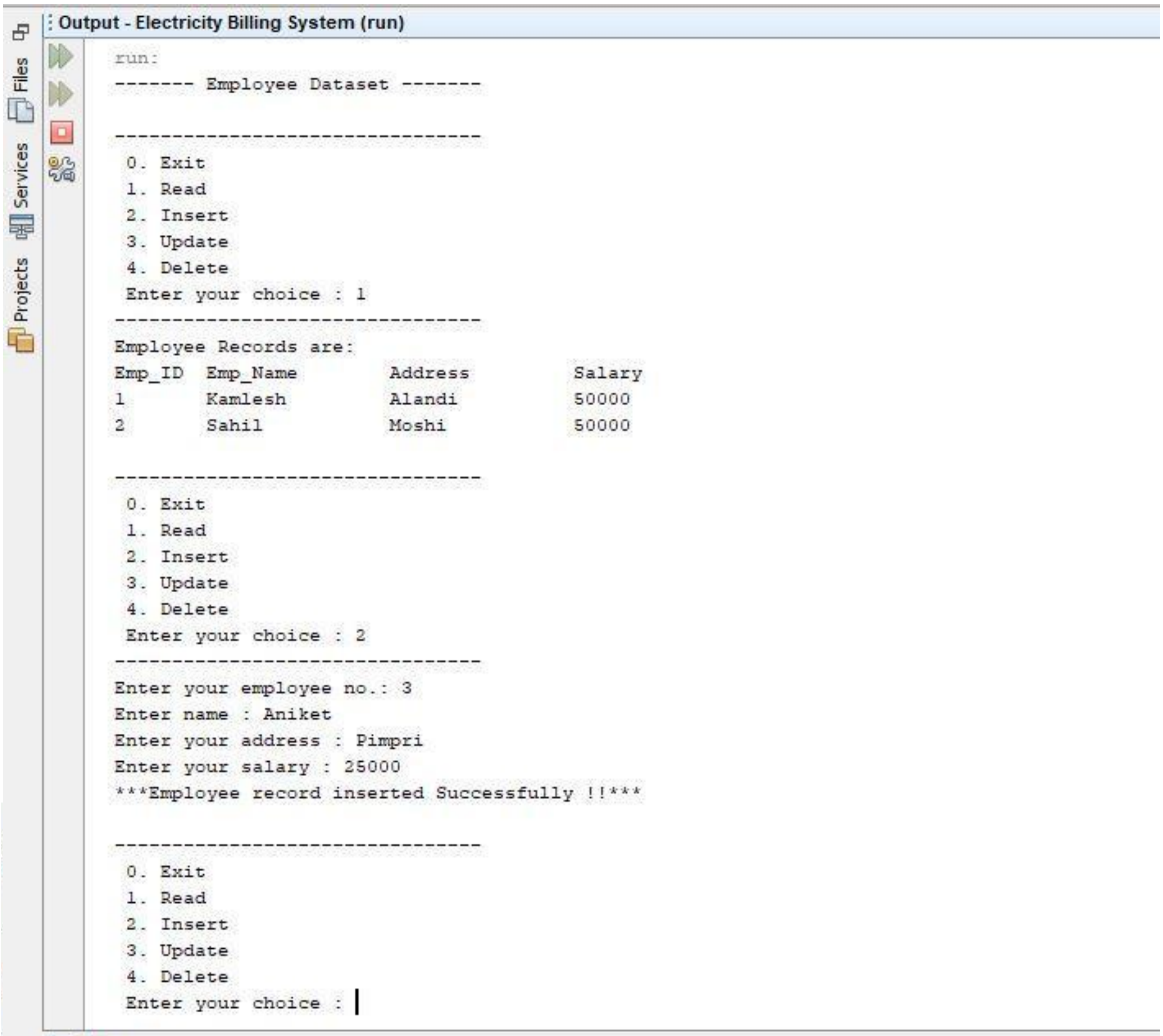
JDBC makes it possible to do establish a connection with a data source, send queries and update statements, and process the results. Simply, JDBC makes it possible to do the following things within a Java application: Establish a connection with a data source. Send queries and update statements to the data source.

Source Code:

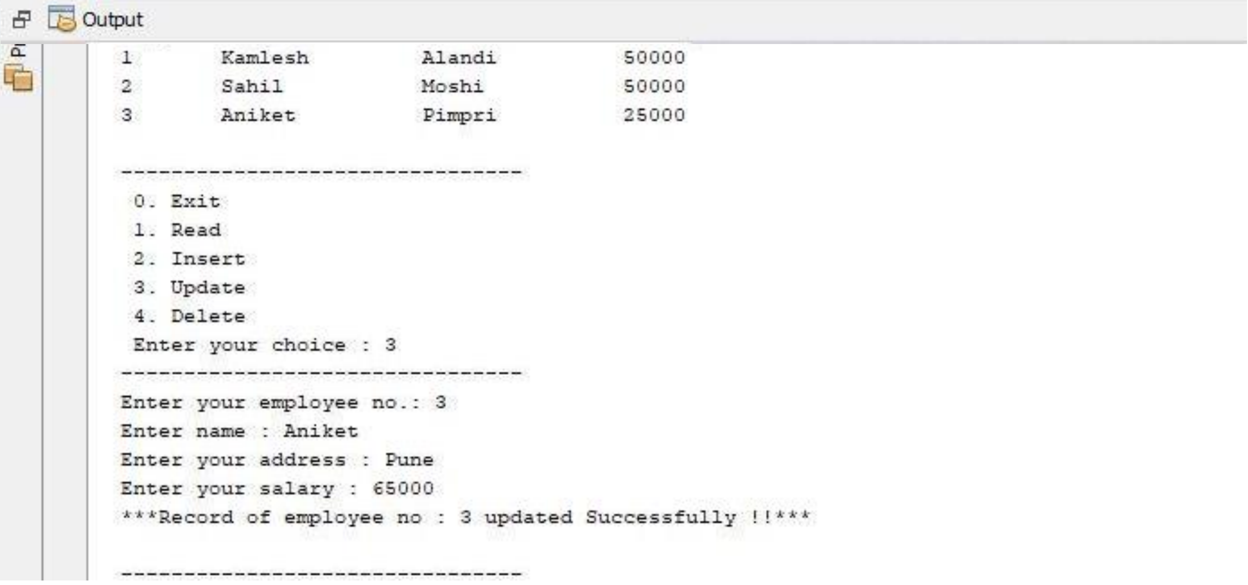
```
package mca1;

import java.sql.*;
import java.util.*;
class Employee{
   Connection c;
   Statement s;
   Scanner sc = new Scanner (System.in);
   public Employee(){
      try{
         Class.forName("com.mysql.cj.jdbc.Driver");
         c =DriverManager.getConnection("jdbc:mysql:///mca1","root","");
         s =c.createStatement();
      }catch(Exception e){
         System.out.println(e);
      }
   }
   public void read(){
      System.out.println("Employee Records are: ");
      System.out.println("Emp_ID\tEmp_Name\tAddress\t\tSalary");
      try{
         PreparedStatement ps=c.prepareStatement("select * from Employee");
         ResultSet rs=ps.executeQuery();
         while(rs.next()){
            System.out.println(rs.getInt("emp_no")+"\t"+rs.getString("emp_name")+
                                 "\t\t"+rs.getString("address")+"\t\t"+rs.getInt("salary"));
         }
      }catch(Exception e){
         System.out.println(e);
      }
   }
   public void insert(){
      System.out.print("Enter your employee no.: ");
      int id = sc.nextInt();
```

```
        System.out.print("Enter name : ");
        String name = sc.next();
        System.out.print("Enter your address : ");
        String address = sc.next();
        System.out.print("Enter your salary : ");
        int salary = sc.nextInt();
        try{
            PreparedStatement ps = c.prepareStatement("insert into Employee
values('"+id+"','"+name+"', '"+address+"', '"+salary+"')");
            ps.executeUpdate();
            System.out.println("***Employee record inserted Successfully !!***");
        }catch(Exception e){
            System.out.println(e);
        }
    }
    public void update(){
        System.out.print("Enter your employee no.: ");
        int id = sc.nextInt();
        System.out.print("Enter name : ");
        String name = sc.next();
        System.out.print("Enter your address : ");
        String address = sc.next();
        System.out.print("Enter your salary : ");
        int salary = sc.nextInt();
        try{
            PreparedStatement ps = c.prepareStatement("update Employee set emp_name =
'"+name+"', address = '"+address+"', salary = '"+salary+"' where emp_no = '"+id+"'");
            ps.executeUpdate();
            System.out.println("***Record of employee no : "+id+" updated Successfully
!!***");
        }catch(Exception e){
            System.out.println(e);
        }
    }
    public  void delete(){
        System.out.print("Enter your employee no.: ");
        int id = sc.nextInt();
        try{
            PreparedStatement ps = c.prepareStatement("delete from Employee where emp_no =
'"+id+"'");
            ps.executeUpdate();
            System.out.println("***Record of employee no : "+id+" deleted Successfully !!***");
        }catch(Exception e){
            System.out.println(e);
        }
    }
    public static void main(String args[]){
        Employee e = new Employee();
        Scanner sc = new Scanner(System.in);
        int ch=10;
    System.out.println("------- Employee Dataset -------");
    while(ch!=0){
      System.out.println("\n--------------------------------");
      System.out.print(" 0. Exit\n 1. Read\n 2. Insert\n 3. Update\n 4. Delete\n Enter your choice : ");
      ch = sc.nextInt();
      System.out.println("--------------------------------");
      switch(ch){
        case 1:
          e.read();
          break;
        case 2:
          e.insert();
          break;
        case 3:
          e.update();
          break;
        case 4:
          e.delete();
          break;
        case 0:
          System.exit(0);
        default:
          System.out.println("Please enter correct option !!");
      }
    }
  }
}
```

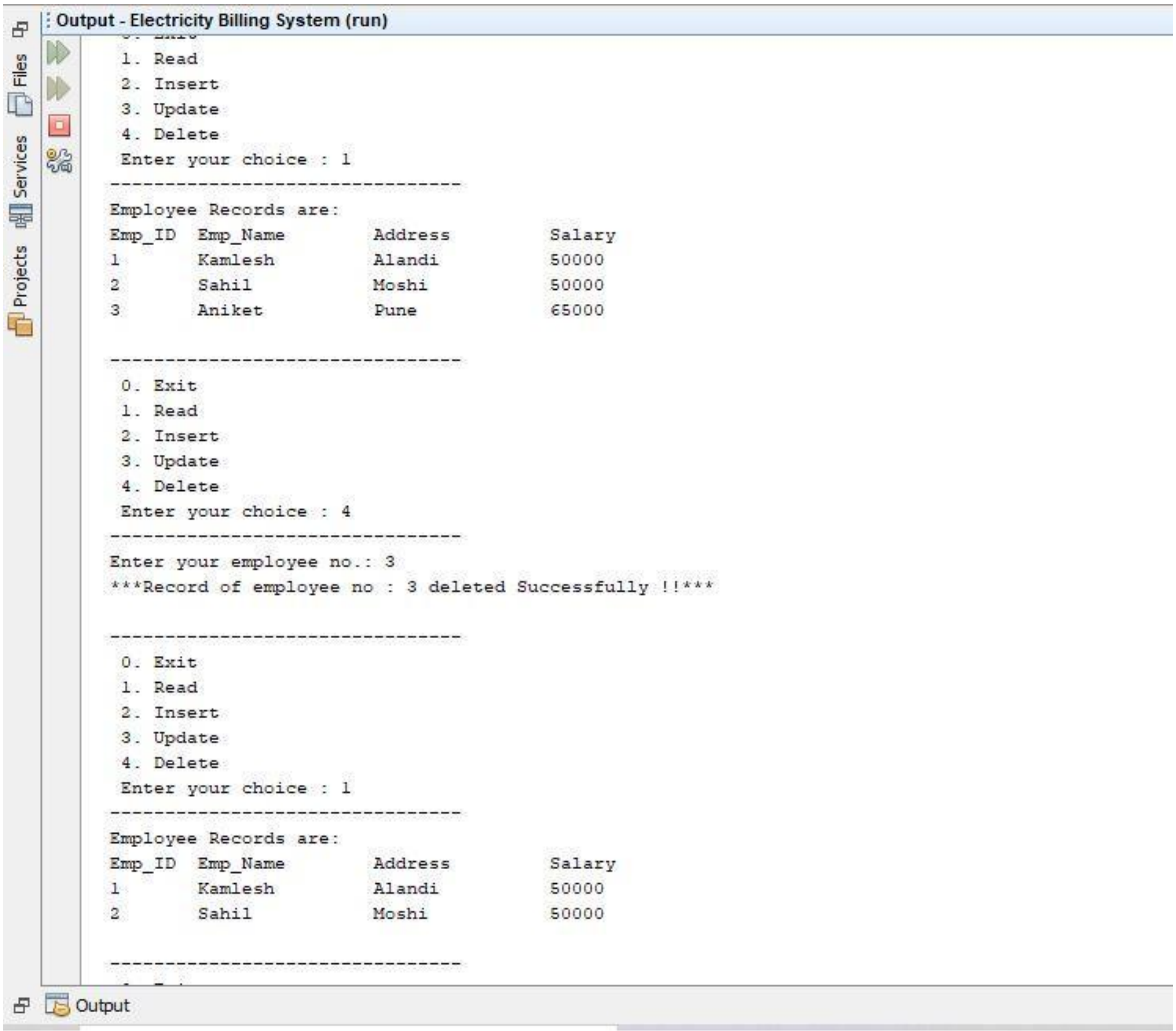
Output:



```
run:
------- Employee Dataset -------

--------------------------------
 0. Exit
 1. Read
 2. Insert
 3. Update
 4. Delete
 Enter your choice : 1
--------------------------------
Employee Records are:
Emp_ID  Emp_Name        Address         Salary
1       Kamlesh         Alandi          50000
2       Sahil           Moshi           50000

--------------------------------
 0. Exit
 1. Read
 2. Insert
 3. Update
 4. Delete
 Enter your choice : 2
--------------------------------
Enter your employee no.: 3
Enter name : Aniket
Enter your address : Pimpri
Enter your salary : 25000
***Employee record inserted Successfully !!***

--------------------------------
 0. Exit
 1. Read
 2. Insert
 3. Update
 4. Delete
 Enter your choice : 
```



```
1       Kamlesh         Alandi          50000
2       Sahil           Moshi           50000
3       Aniket          Pimpri          25000

--------------------------------
 0. Exit
 1. Read
 2. Insert
 3. Update
 4. Delete
 Enter your choice : 3
--------------------------------
Enter your employee no.: 3
Enter name : Aniket
Enter your address : Pune
Enter your salary : 65000
***Record of employee no : 3 updated Successfully !!***

--------------------------------
```



```
 1. Read
 2. Insert
 3. Update
 4. Delete
 Enter your choice : 1
--------------------------------
Employee Records are:
Emp_ID  Emp_Name        Address         Salary
1       Kamlesh         Alandi          50000
2       Sahil           Moshi           50000
3       Aniket          Pune            65000

--------------------------------
 0. Exit
 1. Read
 2. Insert
 3. Update
 4. Delete
 Enter your choice : 4
--------------------------------
Enter your employee no.: 3
***Record of employee no : 3 deleted Successfully !!***

--------------------------------
 0. Exit
 1. Read
 2. Insert
 3. Update
 4. Delete
 Enter your choice : 1
--------------------------------
Employee Records are:
Emp_ID  Emp_Name        Address         Salary
1       Kamlesh         Alandi          50000
2       Sahil           Moshi           50000

--------------------------------
```

Roll No.22239

Assignment 19: Write a Java Servlet Application for login page with proper validations.

Summary:

Servlets are the Java programs that runs on the Java-enabled web server or application server. They are used to handle the request obtained from the web server, process the request, produce the response, and then send response back to the web server. Properties of Servlets: Servlets work on the server-side.

Source Code:

Login1>login.html

```
<html>
   <body>
      <form action=http://localhost:8080/Login1/Login>
         Email: <input type=email name=email required>
         <br>
         Password: <input type=password name=pass required><br>
         <input type=submit value=LOGIN>
      </form>
      <p>correct Email: abcd@gmail.com<br>password:1234
   </body
</html>
```

Login1> WEB-INF>web.xml

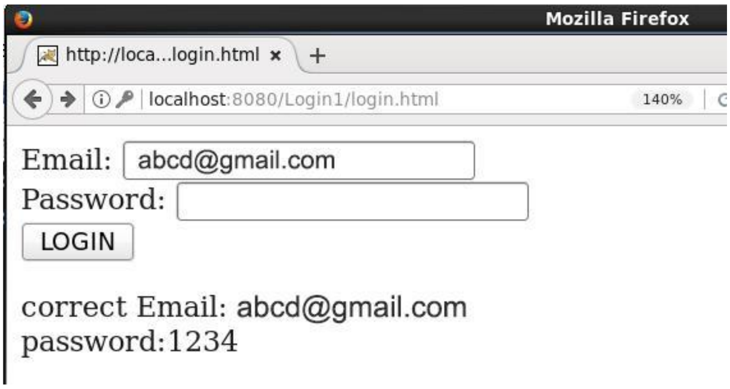
```
<?xml version="1.0" ?>
<web-app>
   <servlet>
      <servlet-name>Login</servlet-name>
      <servlet-class>Login</servlet-class>
   </servlet>

   <servlet-mapping>
      <servlet-name>Login</servlet-name>
      <url-pattern>/Login</url-pattern>
   </servlet-mapping>
</web-app>
```

Login1> WEB-INF>Classes>Login.java

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.io.*;

public class Login extends HttpServlet{
    public void doGet(HttpServletRequest request,HttpServletResponse response){
        try{
        PrintWriter out= response.getWriter();
        out.println("<html><body>");

        String email = request.getParameter("email");
        String pass = request.getParameter("pass");
        out.println("<p>You entered:<br>Email: "+email+"<br>Password: "+pass+"</p>");

        if(email.equals("abcd@gmail.com") && pass.equals("1234")){
            out.println("<p>Login Successfull!!</p>");
        }
        else{
            out.println("<p>Incorrect LoginId or Password!!</p>");
        }
            out.println("</html></body>");
        }
        catch(Exception e){
            System.out.println("Exception : "+e);
        }
    }
}
```

Output:

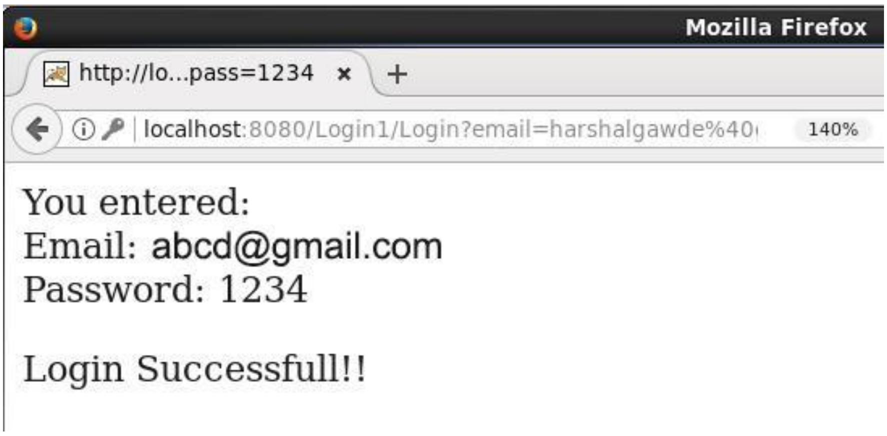
1. Login Form



Email: abcd@gmail.com
Password:

correct Email: abcd@gmail.com
password:1234

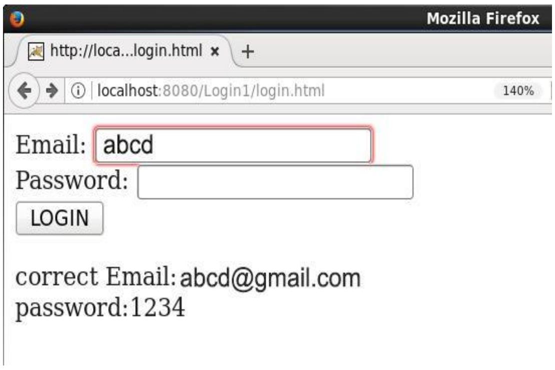
2. After entering correct ID and Password



You entered:
Email: abcd@gmail.com
Password: 1234

Login Successfull!!

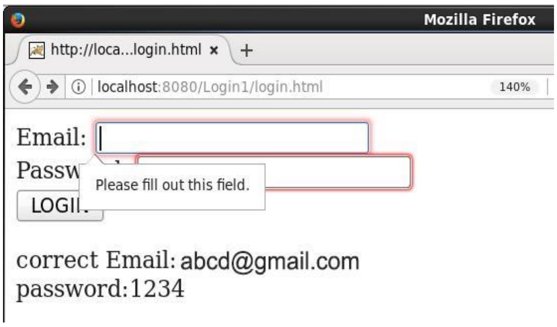
3. Validation



Email: abcd
Password:
LOGIN

correct Email: abcd@gmail.com
password:1234

4. Validation



Email:
Passw
Please fill out this field.
LOGI

correct Email: abcd@gmail.com
password:1234

5. Incorrect ID/Password

Roll No.22239

Assignment 20: Write a Java program to design Registration page using JSP.

Summary:

JSP is generally used in the front end or GUI layer to create views, while Servlet is mostly used in the backend as Controller in MVC pattern whose job is to capture and redirect HTTP request for further processing. In short, a Servlet is HTML in Java, while a JSP is Java in HTML.

Source Code:

Register> register.html

```
<html>
   <body>
      <form action=http://localhost:8080/Register/register.jsp>
      <h2>REGISTERATION PAGE:</h2><br>
         Email: <input type=email name=email required>
         <br>
         Password: <input type=password name=pass required><br>
         <input type=submit value=LOGIN>
      </form>
   </body
</html>
```

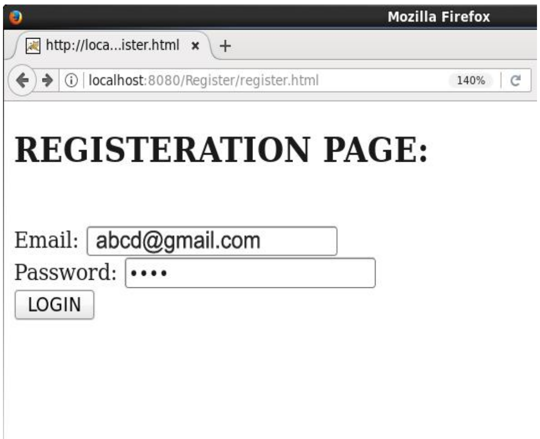
Register> register.jsp

```
<%@page import="java.util.*,java.text.*,java.sql.*" %>
<%
     String email = request.getParameter("email");
     String pass = request.getParameter("pass");
               try{
                         Connection con;
                         Statement st;
                         ResultSet rs;

                         Class.forName("org.postgresql.Driver");
                         con=DriverManager.getConnection("jdbc:postgresql:te512","postgres","");

                         st=con.createStatement();

                         rs= st.executeQuery("select * from login where email='"+email+"'");

                         if(!rs.next()){  //if no such email is already present
```

```
                        st.executeUpdate("insert into login values('"+email+"','"+pass+"')");
                        out.println("Registeration Successfull! ");

                }else{     //if there is already same email in dB
                        out.println("Registeration Failed! <br> The Email entered Already
Exists!!");
                }

                out.println("<br><br>Email:"+email+"<br>Password:"+pass);
        }
        catch(Exception e){
                out.println("Some eXCEPTION Occured"+e);
        }
%>
```

Output:

1. **Enter Email and Password**



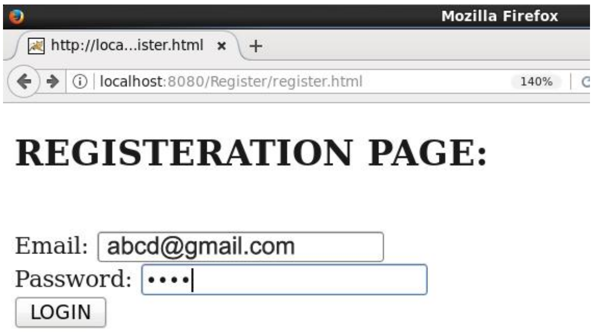
REGISTERATION PAGE:

Email: abcd@gmail.com
Password: ••••
LOGIN

2. **Registration Successful**

3. **Entering already existing mail**

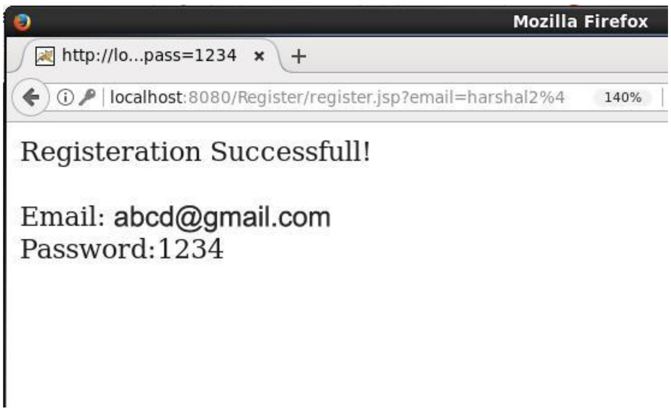
4. **Registration Failed**



REGISTERATION PAGE:

Email: abcd@gmail.com
Password: ••••

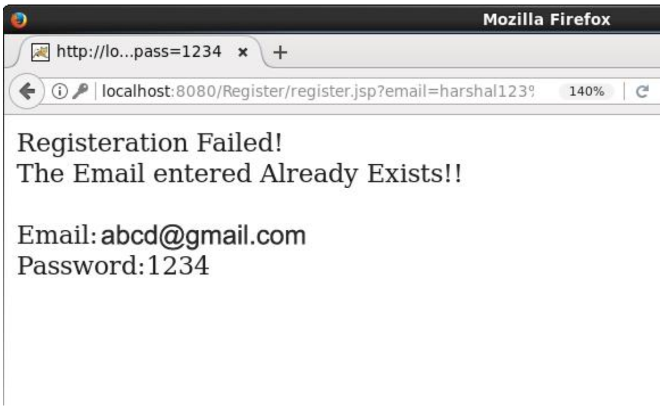
LOGIN



Registeration Successfull!

Email: abcd@gmail.com
Password:1234

INDEX



Registeration Failed!
The Email entered Already Exists!!

Email:abcd@gmail.com
Password:1234