Advanced Java Programming

Lab Programs – 2019-20

Please Note:

- This document contains all AJP lab programs along with sample output.
- The programs are taken the official Lab Manual created by Mrs. Rajalaxmi Hegde, Assistant Professor Grade-II. Later small modifications are done.
- The lab programs in this document are not to be considered as final.
- The will give the proper output and these are only for reference.
- The document is not an official copy. Creator of this document is not responsible if you don’t get the proper output for the programs.
- Please execute these and test by yourself. If any mistakes are found those will be modified and new copy will be updated to google drive “V Semester Study Materials”.

Prepared by:
Shawn Linton Miranda
4NM17CS164

Index

PART - A

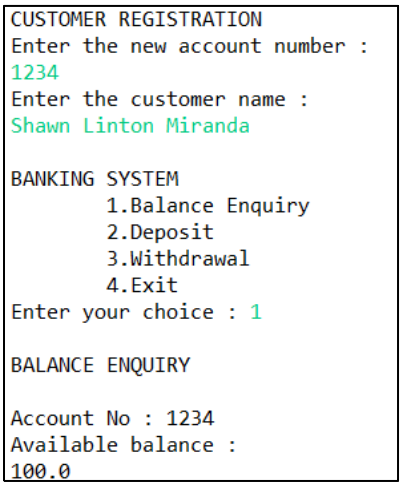
1. Bank Transactions

Create a class customer for a bank. User inheritance, construction for base and derived class. Use default arguments and super keyword. Implement the methods deposit( ), withdraw( ) and getBalance( ).

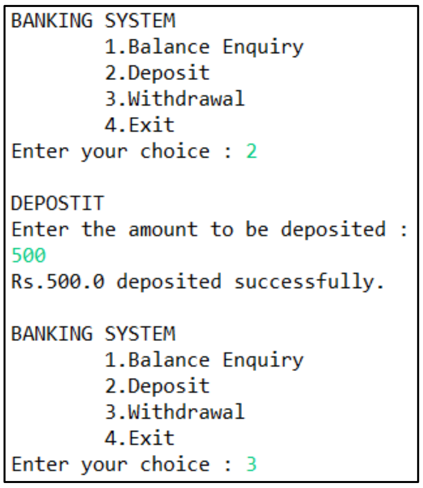
Program:

```
import java.util.Scanner;
class Customer
{
        long acc_no;
        String cust_name;
        Customer(long acc_no,String cust_name)
        {
                this.acc_no=acc_no;
                this.cust_name=cust_name;
        }
}

class Transaction extends Customer
{
        double amt;
        double balance;
        Transaction(long acc_no,String cust_name, double balance)
        {
                super(acc_no,cust_name);
                this.balance=balance;
        }

        void deposit()
        {
                Scanner sc=new Scanner(System.in);
                System.out.println("\nDEPOSTIT");
                System.out.println("Enter the amount to be deposited : ");
                amt=sc.nextDouble();
                balance+=amt;
                System.out.println("Rs."+amt+" deposited successfully.");
        }

        void withdrawal()
        {
                Scanner sc=new Scanner(System.in);
                System.out.println("\nWITHDRAWAL");
                System.out.println("Enter the required amount : ");
                amt=sc.nextDouble();
                if(amt>balance) {
                        System.out.println("Sorry! Insufficient balance.");
                }
                else
                {
                        balance-=amt;
                        System.out.println("Rs."+amt+" has been debited from your account.");
                }
        }
        void balanceEnquiry()
        {
                System.out.println("\nBALANCE ENQUIRY");
                System.out.println("\nAccount No : "+super.acc_no+"\nAvailable balance : \n"+balance+"\n");
        }
}
public class BankingSystem {
        public static void main(String args[])
        {
                boolean control=true;
                Scanner sc=new Scanner(System.in);
                System.out.println("CUSTOMER REGISTRATION");
                System.out.println("Enter the new account number :");
                long acc_no=sc.nextLong();
                sc.nextLine();    //To consume the \n left by previous scan
                System.out.println("Enter the customer name :");
                String cust_name=sc.nextLine();
                Transaction acc=new Transaction(acc_no,cust_name,100.00);
                while(control) {
                        System.out.print("\nBANKING SYSTEM\n\t1.Balance
                        Enquiry\n\t2.Deposit\n\t3.Withdrawal\n\t4.Exit\nEnter your choice : ");
                        int ch=sc.nextInt();
                        switch(ch)
                        {
                                case 1:acc.balanceEnquiry();
                                break;
                                case 2:acc.deposit();
                                break;
                                case 3:acc.withdrawal();
                                break;
                                case 4:control=false;
                                break;
                                default:System.out.println("Invalid choice!");
                        }
                }
                System.out.println("Thank you.");
        }
  }
```

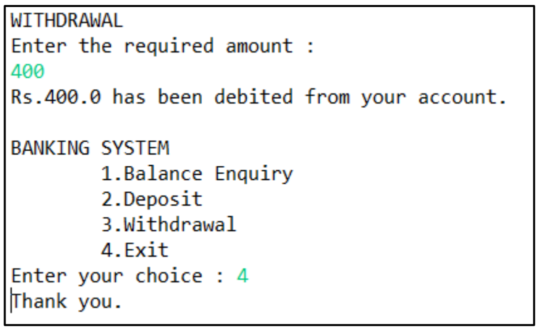
Output:



```
CUSTOMER REGISTRATION
Enter the new account number :
1234
Enter the customer name :
Shawn Linton Miranda

BANKING SYSTEM
        1.Balance Enquiry
        2.Deposit
        3.Withdrawal
        4.Exit
Enter your choice : 1

BALANCE ENQUIRY

Account No : 1234
Available balance :
100.0
```



```
BANKING SYSTEM
        1.Balance Enquiry
        2.Deposit
        3.Withdrawal
        4.Exit
Enter your choice : 2

DEPOSTIT
Enter the amount to be deposited :
500
Rs.500.0 deposited successfully.

BANKING SYSTEM
        1.Balance Enquiry
        2.Deposit
        3.Withdrawal
        4.Exit
Enter your choice : 3
```



```
WITHDRAWAL
Enter the required amount :
400
Rs.400.0 has been debited from your account.

BANKING SYSTEM
        1.Balance Enquiry
        2.Deposit
        3.Withdrawal
        4.Exit
Enter your choice : 4
Thank you.
```

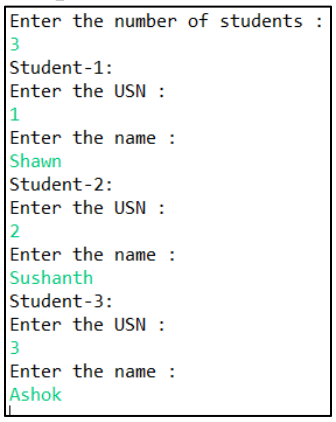
2. *Array Index Out Of Bounds Exception Demonstration*

Write a java program to display names and roll number of students. Initialize details of students. Handle ArrayIndexOutOfBoundsException to avoid illegal termination of program.

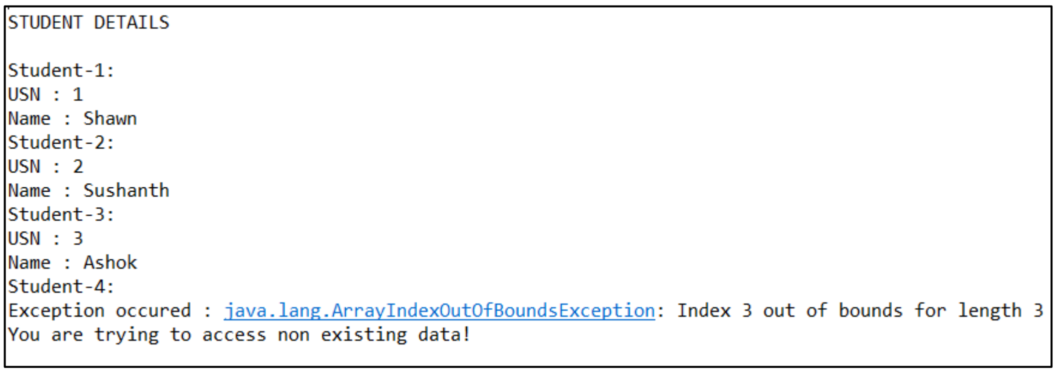
Program:

```
import java.util.Scanner;
class Students  {
        int usn;
        String name;
        Students(int usn, String name)  {
                this.usn=usn;
                this.name=name;
        }
}
public class StudentDetails  {
        int n;
        Students stud[];
        Scanner sc=new Scanner(System.in);
        public static void main(String args[])  {
                StudentDetails obj=new StudentDetails();
                obj.readData();
                obj.displayData();
        }
        public void readData()
        {
                System.out.println("Enter the number of students : ");
                n=sc.nextInt();
                stud=new Students[n];
                for(int i=0; i<n; i++)  {
                        System.out.println("Student-"+(i+1)+":\nEnter the USN : ");
                        int usn=sc.nextInt();
                        System.out.println("Enter the name : ");
                        sc.nextLine();
                        String name=sc.nextLine();
                        stud[i]=new Students(usn,name);
                }
                sc.close();
        }
        public void displayData()  {
                System.out.println("\nSTUDENT DETAILS\n");
                for(int i=0; i<n+1; i++)  {
                        try  {
                                System.out.println("Student-"+(i+1)+":");
                                System.out.println("USN : "+stud[i].usn);
                                System.out.println("Name : "+stud[i].name);
                        }
                        catch(ArrayIndexOutOfBoundsException e)  {
                                System.out.println("Exception occured : "+e);
                                System.out.println("You are trying to access non existing data!");
                        }
                }
        }
}
```

Output:



```
Enter the number of students :
3
Student-1:
Enter the USN :
1
Enter the name :
Shawn
Student-2:
Enter the USN :
2
Enter the name :
Sushanth
Student-3:
Enter the USN :
3
Enter the name :
Ashok
```



```
STUDENT DETAILS

Student-1:
USN : 1
Name : Shawn
Student-2:
USN : 2
Name : Sushanth
Student-3:
USN : 3
Name : Ashok
Student-4:
Exception occured : java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3
You are trying to access non existing data!
```

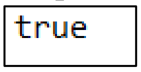
3. Regular Expressions

a. Writer a java program using regular expressions for single occurrence of a patterns in a string.

Program:

```
import java.util.regex.*;
public class PatternMatching1 {
        public static void main(String args[])
        {
                String content="This is book very good.";
                String pattern=".*book.*";
                boolean isMatch=Pattern.matches(pattern, content);
                System.out.println(isMatch);
        }
}
```

Output:



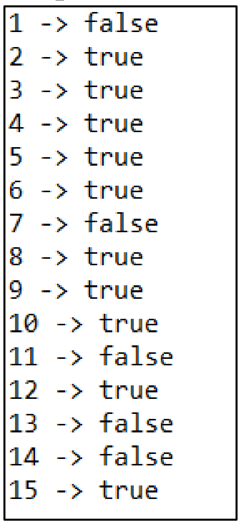
```
true
```

b. Writer a java program using regular expressions for multiple occurrence of a patterns in a string.

Program:

```
import java.util.regex.*;
public class PatternMatching2 {
    public static void main(String args[])
    {
      System.out.println("1 -> "+Pattern.matches("tom","Tom"));
      System.out.println("2 -> "+Pattern.matches("[Tt]om", "Tom"));
      System.out.println("3 -> "+Pattern.matches("[Tt]om", "tom"));
      System.out.println("4 -> "+Pattern.matches("[tT]im|[jJ]in","Tim"));
      System.out.println("5 -> "+Pattern.matches("[tT]im|[jJ]in", "jin"));
      System.out.println("6 -> "+Pattern.matches(".*abc.*", "deabcpq"));
      System.out.println("7 -> "+Pattern.matches("^[^\\d].*","123abc"));
      System.out.println("8 -> "+Pattern.matches("^[^\\d].*","abc123"));
      System.out.println("9 -> "+Pattern.matches("[a-zA-Z][a-zA-Z][a-zA-Z]", "aPz"));
      System.out.println("10 -> "+Pattern.matches("[a-zA-Z][a-zA-Z][a-zA-Z]","aAA"));
      System.out.println("11 -> "+Pattern.matches("[a-zA-Z][a-zA-Z][a-zA-Z]", "apZx"));
      System.out.println("12 -> "+Pattern.matches("\\D*", "abcde"));
      System.out.println("13 -> "+Pattern.matches("\\D*", "abcde123"));
      System.out.println("14 -> "+Pattern.matches("^This$", "This is Nitte"));
      System.out.println("15 -> "+Pattern.matches("^This$", "This"));
    }
}
```

Output:



```
1 -> false
2 -> true
3 -> true
4 -> true
5 -> true
6 -> true
7 -> false
8 -> true
9 -> true
10 -> true
11 -> false
12 -> true
13 -> false
14 -> false
15 -> true
```

4. Java Bean

Write a simple Java Bean program to display the employee details first name, last name, id and location.

Program:

```
import java.io.Serializable;
import java.util.Scanner;
class Person implements Serializable  {
    private static final long serialVersionUID = 1L;
        Person() { }
        private String first_name,last_name,location;
        private int id;
        public String getFirst_name() {
                return first_name;
        }
        public void setFirst_name(String first_name) {
                this.first_name = first_name;
        }
        public String getLast_name() {
                return last_name;
        }
        public void setLast_name(String last_name) {
                this.last_name = last_name;
        }
        public String getLocation() {
                return location;
        }
        public void setLocation(String location) {
                this.location = location;
        }
        public int getId() {
                return id;
        }
        public void setId(int id) {
                this.id = id;
        }
}
public class JavaBeanDemo   {
        public static void main(String args[])  {
                Person p=new Person();
                Scanner sc=new Scanner(System.in);
                System.out.print("Enter first name : ");
                p.setFirst_name(sc.nextLine());
                System.out.print("Enter last name : ");
                p.setLast_name(sc.nextLine());
                System.out.print("Enter id : ");
                p.setId(sc.nextInt());
                System.out.print("Enter the location : ");
                sc.nextLine();
                p.setLocation(sc.nextLine());
                System.out.println("\nPersonal Details:");
                System.out.println("First Name : "+p.getFirst_name()+"\nLast Name : "+p.getLast_name());
                System.out.println("ID : "+p.getId()+"\nLocation : "+p.getLocation());
                sc.close();
        }
}
```

Output:

```
Enter first name : Shawn
Enter last name : Miranda
Enter id : 123
Enter the location : Siddakatte

Personal Details:
First Name : Shawn
Last Name : Miranda
ID : 123
Location : Siddakatte
```

5. Custom Comparator

Write a java program to implement the custom comparator using TreeSet class to perform sorting operation on employee details based on employee name and salary.

Program:

```
import java.util.Comparator;
import java.util.TreeSet;
class Employee
{
        private String name;
        private float salary;
        public Employee(String n,float sal)  {
                name=n;
                salary=sal;
        }
        public String getName()  {
                return name;
        }
        public void setName(String n)  {
                name=n;
        }
        public float getSalary()  {
                return salary;
        }
        public void setSalary(float sal)  {
                salary=sal;
        }
        public String toString()  {
                return("Name : "+name+"  ---  Salary : "+salary);
        }
}
class NameComparator implements Comparator<Employee>  {
        public int compare(Employee e1,Employee e2)  {
                return(e1.getName().compareTo(e2.getName()));
        }
}
class SalaryComparator implements Comparator<Employee> {
        public int compare(Employee e1,Employee e2)  {
                if(e1.getSalary()>e2.getSalary())
                        return 1;
                else if(e1.getSalary()<e2.getSalary())
                        return -1;
                else
                        return 0;
        }
}
public class ComparatorDemo  {
        public static void main(String args[])  {
                TreeSet<Employee> emp1=new TreeSet<Employee>(new NameComparator());
                emp1.add(new Employee("Shawn L M",100000.00f));
                emp1.add(new Employee("Shrivyas",5000.00f));
                emp1.add(new Employee("Akash",49999.50f));
                emp1.add(new Employee("Manjunath",10.00f));
                emp1.add(new Employee("Kiran",20000.00f));
                System.out.println("\nEmployee Details (Sorted based on name) : ");
                for(Employee employee:emp1)  {
                        System.out.println(employee);
                }
                TreeSet<Employee> emp2=new TreeSet<Employee>(new SalaryComparator());
                emp2.add(new Employee("Shawn L M",100000.00f));
                emp2.add(new Employee("Shrivyas",5000.00f));
                emp2.add(new Employee("Akash",49999.50f));
                emp2.add(new Employee("Manjunath",10.00f));
                emp2.add(new Employee("Kiran",20000.00f));
                System.out.println("\nEmployee Details (Sorted based on Salary) : ");
                for(Employee employee:emp2)  {
                        System.out.println(employee);
                }
        }
}
```

Output:

```
Employee Details (Sorted based on name) :
Name : Akash  ---  Salary : 49999.5
Name : Kiran  ---  Salary : 20000.0
Name : Manjunath  ---  Salary : 10.0
Name : Shawn L M  ---  Salary : 100000.0
Name : Shrivyas  ---  Salary : 5000.0

Employee Details (Sorted based on Salary) :
Name : Manjunath  ---  Salary : 10.0
Name : Shrivyas  ---  Salary : 5000.0
Name : Kiran  ---  Salary : 20000.0
Name : Akash  ---  Salary : 49999.5
Name : Shawn L M  ---  Salary : 100000.0
```

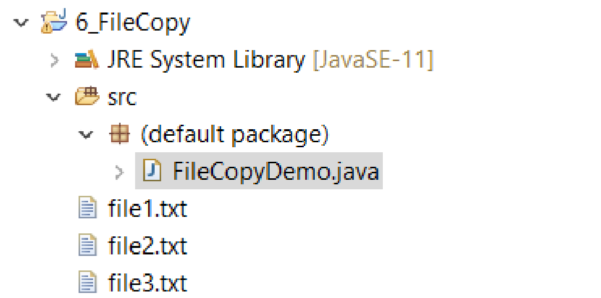
6. Copying file contents

Write a java program to copy contents of two files into third file.

Program:

```
import java.io.*;
public class FileCopyDemo {
        public static void main(String args[]) throws IOException  {
                try  {
                        FileInputStream src1=new FileInputStream("file1.txt");
                        FileInputStream src2=new FileInputStream("file2.txt");
                        FileOutputStream dest=new FileOutputStream("file3.txt");
                        int i;
                        while((i=src1.read())!=-1)  {
                                dest.write(i);
                        }
                        src1.close();
                        while((i=src2.read())!=-1)  {
                                dest.write(i);
                        }
                        src2.close();
                        dest.close();
                }
                catch(IOException e)
                {
```

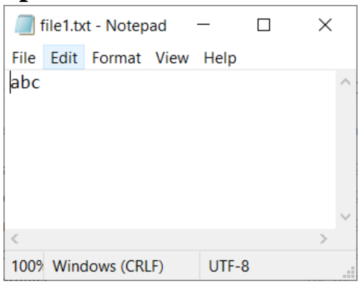
Location of created files:



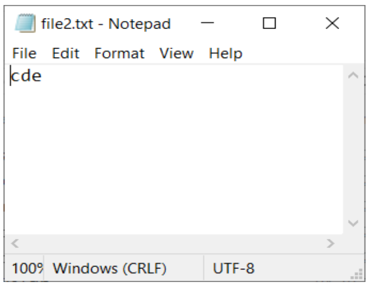
```
6_FileCopy
    JRE System Library [JavaSE-11]
    src
        (default package)
            FileCopyDemo.java
    file1.txt
    file2.txt
    file3.txt
```

```
                        System.out.println("Exception occured : "+e);
                  }
                  System.out.println("Files copied successfully.");
            }
      }
```

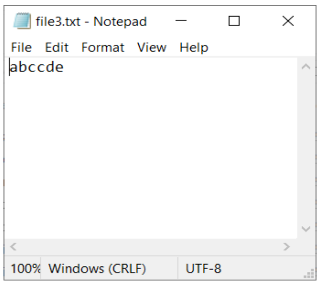
Output:



```
abc
```



```
cde
```



```
abccde
```

7. Sum, Average and Largest of three numbers

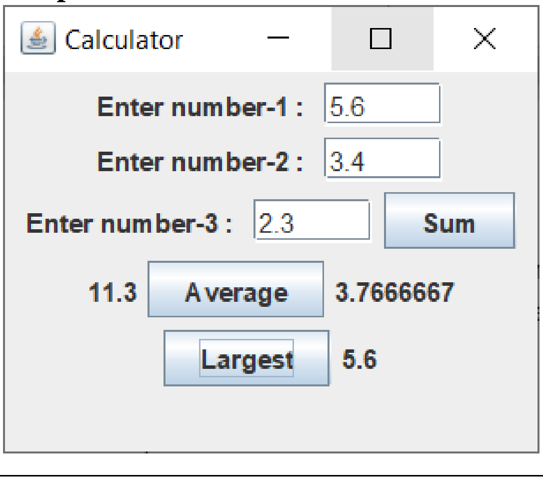
Write a swing program that takes three numbers as input from user and displays sum, average and largest.

Program:

Note:
Some of the lines in this program are written as a comment. Those contain a method setBounds() which is used for the formatting or alignment of the components inside window. If you want to use those please change the Layout specified in the method **setLayout()** to null.
The output shown here is without using setBounds.

```
import java.awt.event.*;
import javax.swing.*;
import java.awt.*;
class Calculator implements ActionListener {
      JFrame jf;
      JLabel l1,l2,l3,sum,avg,largest;
      JButton b1,b2,b3;
      JTextField n1,n2,n3;
      Calculator() {
            jf=new JFrame("Calculator");
            jf.setSize(300, 300);
            jf.setBackground(Color.yellow);
            jf.setLayout(new FlowLayout());
            jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            l1=new JLabel("Enter number-1 : ");
            //l1.setBounds(50,15,100,20);
            jf.add(l1);
            n1=new JTextField(5);
            //n1.setBounds(150,15,100,25);
            jf.add(n1);
            l2=new JLabel("Enter number-2 : ");
            //l2.setBounds(50,45,100,20);
            jf.add(l2);
            n2=new JTextField(5);
            //n2.setBounds(150,45,100,25);
            jf.add(n2);
            l3=new JLabel("Enter number-3 : ");
            //l3.setBounds(50,75,100,20);
            jf.add(l3);
            n3=new JTextField(5);
            //n3.setBounds(150,75,100,25);
            jf.add(n3);
            b1=new JButton("Sum");
            //b1.setBounds(10,150,65,30);
            b1.addActionListener(this);
            jf.add(b1);
            sum=new JLabel("");
            //sum.setBounds(10,190,90,25);
            jf.add(sum);
            b2=new JButton("Average");
            //b2.setBounds(95,150,80,30);
            b2.addActionListener(this);
            jf.add(b2);
            avg=new JLabel("");
            //avg.setBounds(110,190,90,25);
            jf.add(avg);
            b3=new JButton("Largest");
            //b3.setBounds(190,150,80,30);
            b3.addActionListener(this);
            jf.add(b3);
            largest=new JLabel("");
            //largest.setBounds(210,190,80,25);
            jf.add(largest);
            jf.setVisible(true);
        }
        public void actionPerformed(ActionEvent ae)  {
            float num1 = Float.parseFloat(n1.getText());
            float num2 = Float.parseFloat(n2.getText());
            float num3 = Float.parseFloat(n3.getText());
            if(ae.getActionCommand()=="Sum")
            {
                    float res=num1+num2+num3;
                    sum.setText(String.valueOf(res));
            }
            else if(ae.getActionCommand()=="Average")
            {
                    float res=(num1+num2+num3)/3;
                    avg.setText(String.valueOf(res));
            }
            else
            {
                    float lar = Math.max(num1, Math.max(num2, num3));
                    largest.setText(String.valueOf(lar));
            }
        }
}
public class SwingDemo {
        public static void main(String args[])
        {
            SwingUtilities.invokeLater(new Runnable() {
                    public void run()
                    {
                            new Calculator();
                    }
            });
        }
}
```

Output:



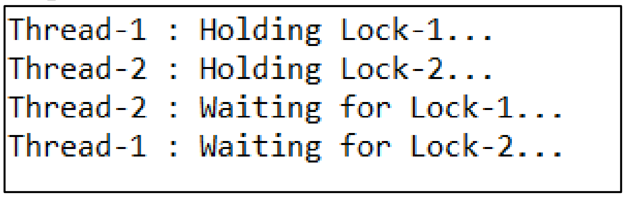
8. Deadlock

Write a java program to demonstrate the appearance of deadlock.

Program:

```java
public class DeadLock  {
    private static Object Lock1=new Object();
    private static Object Lock2=new Object();
    public static void main(String args[])  {
        Thread1 t1=new Thread1();
        Thread2 t2=new Thread2();
        t1.start();
        t2.start();
    }
    private static class Thread1 extends Thread  {
        public void run()  {
            synchronized (Lock1)  {
                System.out.println("Thread-1 : Holding Lock-1...");
                try {
                    Thread.sleep(10);
                }
                catch(InterruptedException e) {}
                System.out.println("Thread-1 : Waiting for Lock-2...");
                synchronized (Lock2)   {
                    System.out.println("Thread-1 : Holding Holding Lock-1 & 2...");
                }
            }
        }
    }
    private static class Thread2 extends Thread {
        public void run()  {
            synchronized(Lock2)  {
                System.out.println("Thread-2 : Holding Lock-2...");
                try {
                    Thread.sleep(10);
                }
                catch(InterruptedException e) {}
                System.out.println("Thread-2 : Waiting for Lock-1...");
                synchronized(Lock1)  {
                    System.out.println("Thread-2 : Holding Lock-1&2...");
                }
            }
        }
    }
}
```

Output:



```
Thread-1 : Holding Lock-1...
Thread-2 : Holding Lock-2...
Thread-2 : Waiting for Lock-1...
Thread-1 : Waiting for Lock-2...
```

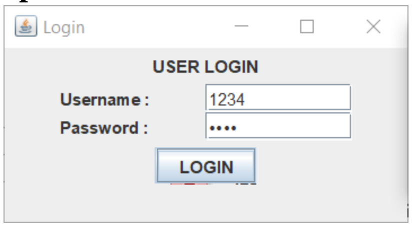
9. *Swing login validation*

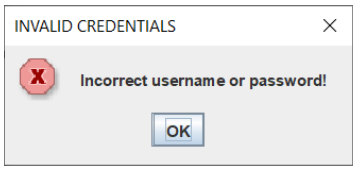
Write a swing program to create a login screen having username and password fields and to go new page based on validation.

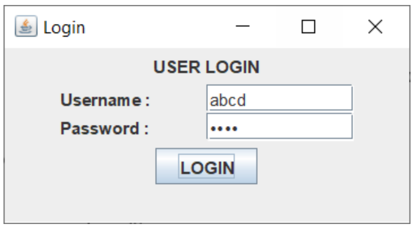
Program:

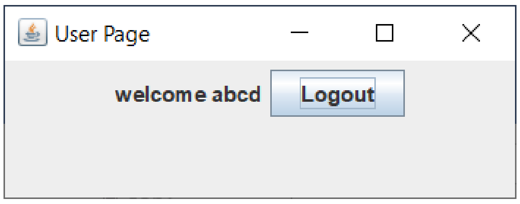
```
import javax.swing.*;           //Login.java
import java.awt.*;
import java.awt.event.*;
class Login implements ActionListener {
        static JFrame jf;
        private JPanel pane;
        private JLabel title,uname,password;
        private JTextField username;
        private JPasswordField pass;
        private JButton login;

        final static private String DEFAULT_USERNAME="abcd";
        final static private String DEFAULT_PASSWORD="1234";

        Login()  {
                jf=new JFrame("Login");
                jf.setSize(300,300);
                jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                jf.setLayout(new FlowLayout());

                title=new JLabel("USER LOGIN");
                jf.add(title);
                //title.setBounds(115, 10, 100, 30);
                uname=new JLabel("Username : ");
                password=new JLabel("Password : ");
                username=new JTextField(10);
                pass=new JPasswordField(10);
                login=new JButton("LOGIN");

                pane=new JPanel(new GridLayout(2,2));
                pane.add(uname);
                pane.add(username);
                pane.add(password);
                pane.add(pass);

                jf.add(pane);
                //pane.setBounds(50, 40, 200, 60);
                jf.add(login);
                //login.setBounds(110, 120, 80, 30);
                login.addActionListener(this);

                jf.setVisible(true);
        }
        public void actionPerformed(ActionEvent ae)  {
                String u=username.getText();
                String p=String.valueOf(pass.getPassword());
                if(u.equals(DEFAULT_USERNAME) && p.equals(DEFAULT_PASSWORD))  {
                        UserPage myPage=new UserPage();
                        JLabel welcome=new JLabel("welcome "+DEFAULT_USERNAME);
                        //welcome.setBounds(20, 50, 150, 30);
                        myPage.add(welcome);
                        JButton logout=new JButton("Logout");
                        //logout.setBounds(200,20,80,30);
                        logout.addActionListener(myPage);
                        myPage.add(logout);
                        myPage.setVisible(true);
                        jf.setVisible(false);
                }
                else
                        JOptionPane.showMessageDialog(jf,"Incorrect username or password!",
                                                        "INVALID CREDENTIALS", JOptionPane.ERROR_MESSAGE);
        }
}
public class LoginPage  {
        public static void main(String args[]) {
                SwingUtilities.invokeLater(new Runnable() {
                        public void run()  {
                                new Login();
                        }
                });
        }
}

import javax.swing.*;   //UserPage.java
import java.awt.*;
import java.awt.event.*;
public class UserPage extends JFrame implements ActionListener{
        UserPage()
        {
                super("User Page");
                setSize(300,300);
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                setLayout(new FlowLayout());
        }
        public void actionPerformed(ActionEvent ae)
        {
                dispose();
                Login.jf.setVisible(true);
        }
}
```

Output:









PART - B

1. User Details using swings

Write a java swing program to get the user details. The form should have TextFields to take name and phone, Radio buttons for selecting gender, List for choosing city, checkboxes for choosing hobbies. Display all the details entered by user in the alert box.

Program:

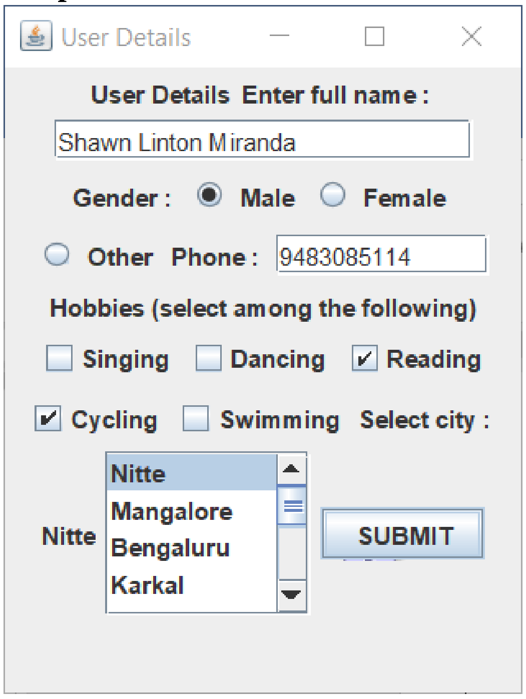
Note:
Some of the lines in this program are written as a comment. Those contain a method setBounds() which is used for the formatting or alignment of the components inside window. If you want to use those please change the Layout specified in the method **setLayout()** to null.
The output shown here is without using setBounds.

```
import javax.swing.*;
import javax.swing.event.*;
import java.awt.*;
import java.awt.event.*;
class Users implements ActionListener,ListSelectionListener {
        JFrame jf;
        JLabel title,uname,phone,gender,hobbies,city,selCity;
        JTextField name,ph;
        JRadioButton male,female,other;
        ButtonGroup bg;
        JList<String> cities;
        JButton submit;
        JCheckBox h1,h2,h3,h4,h5;
        JScrollPane jp;
        String cityList[]= {"Nitte","Mangalore","Bengaluru","Karkal","Delhi","Mumbai"};
        Users()  {
                jf=new JFrame("User Details");
                jf.setSize(500,700);
                jf.setLayout(new FlowLayout());
                jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

                title=new JLabel("User Details");
                //title.setBounds(200,10,100,30);
                jf.add(title);

                uname=new JLabel("Enter full name : ");
                jf.add(uname);
                //uname.setBounds(50,50,200,30);
                name=new JTextField(20);
                jf.add(name);
                //name.setBounds(50,80,200,30);

                gender=new JLabel("Gender : ");
                jf.add(gender);
                //gender.setBounds(50,120,100,30);
                bg=new ButtonGroup();
                male=new JRadioButton(" Male");
                bg.add(male);
                jf.add(male);
                //male.setBounds(50,140,70,30);
                female=new JRadioButton(" Female");
                bg.add(female);
                jf.add(female);
                //female.setBounds(125,140,80,30);
                other=new JRadioButton(" Other");
                bg.add(other);
                jf.add(other);
                //other.setBounds(210,140,80,30);
```

```
		phone=new JLabel("Phone : ");
		jf.add(phone);
		//phone.setBounds(50, 180, 100, 30);
		ph=new JTextField(10);
		jf.add(ph);
		//ph.setBounds(50,210,150,30);

		hobbies=new JLabel("Hobbies (select among the following)");
		jf.add(hobbies);
		//hobbies.setBounds(50, 250, 300, 30);
		h1=new JCheckBox("Singing");
		jf.add(h1);
		//h1.setBounds(50,280,100,30);
		h2=new JCheckBox("Dancing");
		jf.add(h2);
		//h2.setBounds(50,310,100,30);
		h3=new JCheckBox("Reading");
		jf.add(h3);
		//h3.setBounds(50,340,100,30);
		h4=new JCheckBox("Cycling");
		jf.add(h4);
		//h4.setBounds(50,370,100,30);
		h5=new JCheckBox("Swimming");
		jf.add(h5);
		//h5.setBounds(50,400,100,30);

		city=new JLabel("Select city : ");
		jf.add(city);
		//city.setBounds(50,450,100,30);
		selCity=new JLabel("");
		jf.add(selCity);
		selCity.setVisible(false);
		//selCity.setBounds(133,450,200,30);
		cities=new JList<String>(cityList);
		jp=new JScrollPane(cities);
		jp.setPreferredSize(new Dimension(100,80));
		//jp.setBounds(50,480,200,80);
		jf.add(jp);
		cities.addListSelectionListener(this);

		submit=new JButton("SUBMIT");
		jf.add(submit);
		//submit.setBounds(210,580,100,30);
		submit.addActionListener(this);
		jf.setVisible(true);
	}
	public void actionPerformed(ActionEvent ae)  {
		String result="";
		String n=name.getText();
		String pno=ph.getText();
		String c=selCity.getText();
		String gen="";
		if(male.isSelected()) gen="Male";
		else if(female.isSelected()) gen="Female";
		else if(other.isSelected()) gen="Other";
```

```
                String h="";
                if(h1.isSelected()) h+=h1.getActionCommand();
                if (h2.isSelected())  {
                        if(!h.equals("")) h+=", ";
                        h+=h2.getActionCommand();
                }
                if (h3.isSelected())   {
                        if(!h.equals("")) h+=", ";
                        h+=h3.getActionCommand();
                }
                if (h4.isSelected())  {
                        if(!h.equals("")) h+=", ";
                        h+=h4.getActionCommand();
                }
                if (h5.isSelected())   {
                        if(!h.equals("")) h+=", ";
                        h+=h5.getActionCommand();
                }

                result+="Name : "+n+"\nGender : "+gen+"\nPhone : "+pno+"\nCity : "+c+"\nHobbies : "+h;
                JOptionPane.showMessageDialog(jf, "USER DETAILS: \n"+result);
                selCity.setVisible(false);
        }
        public void valueChanged(ListSelectionEvent le)
        {
                int i=cities.getSelectedIndex();
                if(i!=-1)  {
                        selCity.setVisible(true);
                        selCity.setText(cityList[i]);
                }
        }
}

public class UserSwingDemo   {
        public static void main(String args[])       {
                SwingUtilities.invokeLater(new Runnable() {
                        public void run() {
                                new Users();
                        }
                });
        }
}
```

Output:



User Details

User Details Enter full name : Shawn Linton Miranda

Gender : [✓ Male] Female [ ] Other Phone : 9483085114

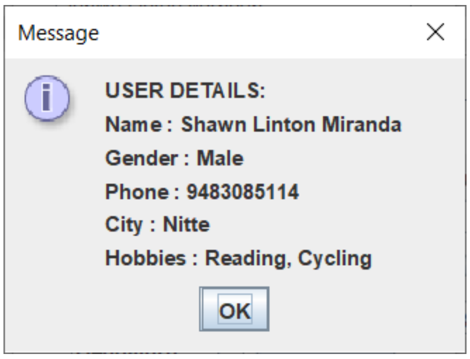
Hobbies (select among the following)

Singing [ ] Dancing [ ] Reading [✓] Cycling [✓] Swimming [ ] Select city :

Nitte

[✓ Nitte] Mangalore Bengaluru Karkal

SUBMIT



```
Message

USER DETAILS:
Name : Shawn Linton Miranda
Gender : Male
Phone : 9483085114
City : Nitte
Hobbies : Reading, Cycling

OK
```

2. Random Access File

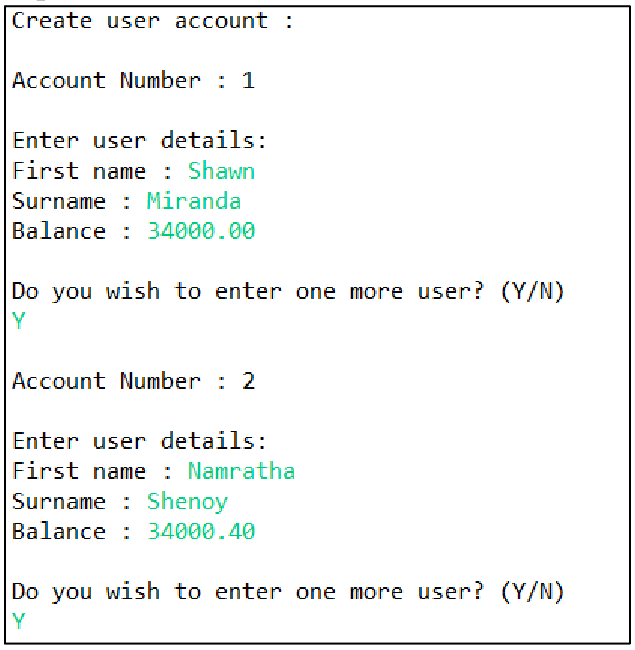
Write a java program to performs these operation on a Random access file containing 3 fields namely

Program:

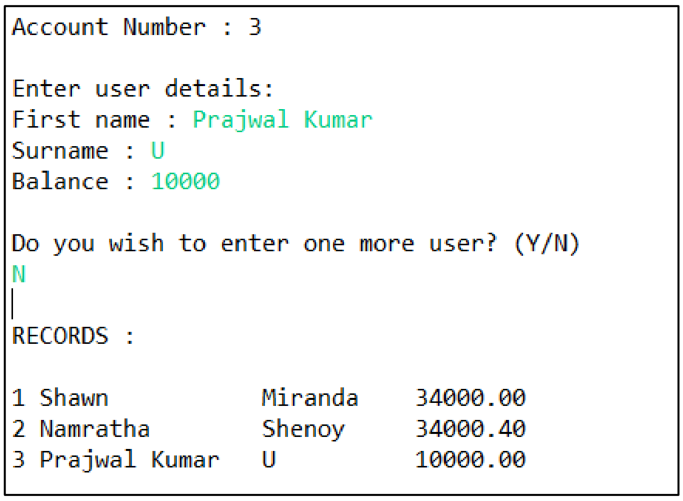
```
import java.io.IOException;
import java.util.Scanner;
import java.io.RandomAccessFile;

public class RandomAccessDemo {
        private static final int REC_SIZE = 62;
        private static final int NAME_SIZE = 15;
        private static final int SURNAME_SIZE = 10;
        private static long accNum=0;
        private static String name,surname;
        private static float balance;
        public static void main(String args[]) throws IOException   {
                RandomAccessFile account=new RandomAccessFile("accounts.txt","rw");
                Scanner sc=new Scanner(System.in);
                String reply;
                System.out.println("Create user account : ");
                do  {
                        accNum++;
                        System.out.println("\nAccount Number : "+accNum+"\n\nEnter user details:");
                        System.out.print("First name : ");
                        name=sc.nextLine();
                        System.out.print("Surname : ");
                        surname=sc.nextLine();
                        System.out.print("Balance : ");
                        balance=sc.nextFloat();
                        sc.nextLine();
                        writeRecord(account);
                        System.out.println("\nDo you wish to enter one more user? (Y/N)");
                        reply=sc.nextLine();
                }while(reply.equals("Y") || reply.equals("y"));
                System.out.println("\nRECORDS : \n");
                showRecords(account);
                sc.close();
        }
        public static void writeRecord(RandomAccessFile file) throws IOException   {
                long filePos=(accNum-1)*REC_SIZE;
                file.seek(filePos);
                file.writeLong(accNum);
                writeString(file,name,NAME_SIZE);
                writeString(file,surname,SURNAME_SIZE);
                file.writeFloat(balance);
        }
        public static void writeString(RandomAccessFile file,String text, int fixedSize) throws IOException  {
                int size=text.length();
                if(size<=fixedSize)  {
                        file.writeChars(text);
                        for(int i=size; i<fixedSize; i++)
                                file.writeChar(' ');
                }
```

```
            else
                    file.writeChars(text.substring(0,fixedSize));
        }
        public static void showRecords(RandomAccessFile file) throws IOException  {
            long numRecords=file.length()/REC_SIZE;
            file.seek(0);
            for(int i=0; i<numRecords; i++)   {
                    accNum=file.readLong();
                    name=readString(file,NAME_SIZE);
                    surname=readString(file,SURNAME_SIZE);
                    balance=file.readFloat();
                    System.out.printf("" + accNum+ " " + name+ " " + surname + " "+ "%.2f \n",balance);
            }
        }
        public static String readString(RandomAccessFile file,int fixedSize) throws IOException  {
            String value="";
            for(int i=0;i<fixedSize; i++)
                    value+=file.readChar();
            return value;
        }
  }
```

Output:



```
Create user account :

Account Number : 1

Enter user details:
First name : Shawn
Surname : Miranda
Balance : 34000.00

Do you wish to enter one more user? (Y/N)
Y

Account Number : 2

Enter user details:
First name : Namratha
Surname : Shenoy
Balance : 34000.40

Do you wish to enter one more user? (Y/N)
Y
```



```
Account Number : 3

Enter user details:
First name : Prajwal Kumar
Surname : U
Balance : 10000

Do you wish to enter one more user? (Y/N)
N
|
RECORDS :

1 Shawn         Miranda    34000.00
2 Namratha      Shenoy     34000.40
3 Prajwal Kumar U          10000.00
```

3. JDBC – Bank Account

Write a JDBC program to perform following operations on database table Accounts (accno, name, balance)
1.Insert a record   2.Delete a record   3.Display all records.

Program:

```
  import java.sql.*;
  import java.util.Scanner;
  class Bank  {
        private static Connection con;
        private static Statement stmt;
        private static ResultSet rs;
        private static Scanner sc;
```

Note:

- To connect with MySQL (Wamp / Xampp server) Use :

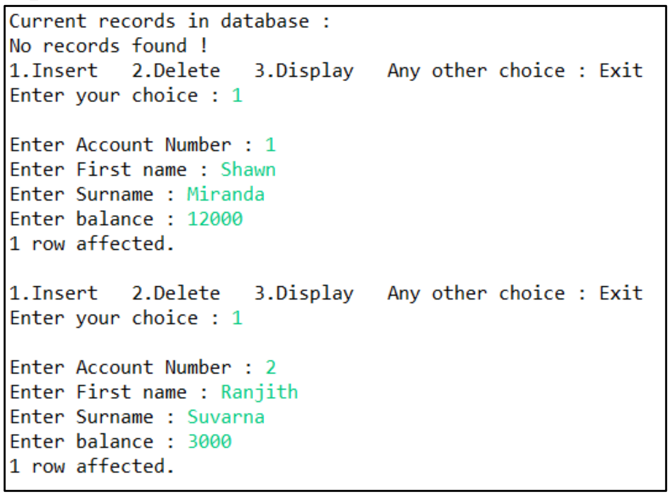
```
com.mysql.jdbc.Driver
"jdbc:mysql://localhost:3306/Student?useSSL=false","username","pass"
```

- To connect with MS SQL server use :

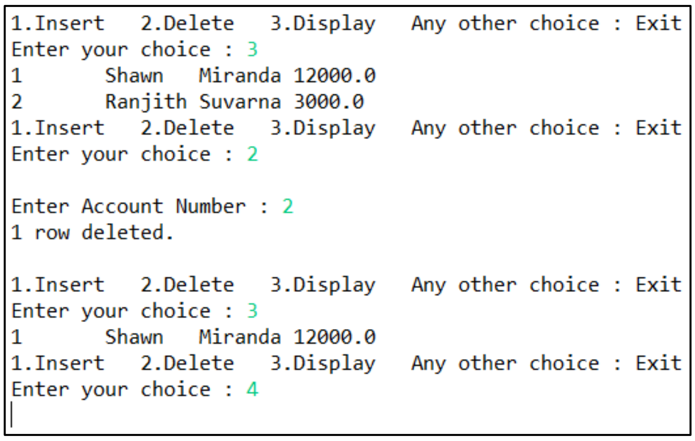
```
com.microsoft.sqlserver.jdbc.SQLServerDriver
"jdbc:sqlserver://server_name_of_ssms","username","password"
After this include stmt.executeUpdate(“USE db_name”);
```

```
        Bank() throws SQLException  {
            try {
                            Class.forName("com.mysql.jdbc.Driver ");
            }
            catch(ClassNotFoundException e)   {
                            System.out.println("Unable to load the driver!!");
                            System.exit(1);
            }
            try {
                            con=DriverManager.getConnection("jdbc:mysql://172.16.2.3/student ","student","student");
            }
            catch(SQLException e)   {
                            System.out.println("Cannot connect to the database !");
                            System.exit(1);
            }
            stmt=con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);
            sc=new Scanner(System.in);
        }

        void insertRow() throws SQLException    {
            System.out.print("\nEnter Account Number : ");
            int accno=sc.nextInt();
            sc.nextLine();
            System.out.print("Enter First name : ");
            String fname=sc.nextLine();
            System.out.print("Enter Surname : ");
            String surname=sc.nextLine();
            System.out.print("Enter balance : ");
            Float balance=sc.nextFloat();
            String insert="INSERT INTO Bank VALUES("+accno+",'"+fname+"','"+surname+"',"+balance+");";
            int i=stmt.executeUpdate(insert);
            if(i>0) System.out.println("1 row affected.\n");
        }

        void deleteRow() throws SQLException   {
            sc=new Scanner(System.in);
            System.out.print("\nEnter Account Number : ");
            int accno=sc.nextInt();
            String delete="DELETE FROM Bank WHERE accno="+accno;
            int i=stmt.executeUpdate(delete);
            if(i>0) System.out.println("1 row deleted.\n");
        }

        void displayData() throws SQLException
        {
            String select="SELECT * FROM Bank";
            rs=stmt.executeQuery(select);
            int count=0;
            while(rs.next())
            {
                    count++;
                    System.out.println(rs.getInt(1)+"\t"+rs.getString(2)+"\t"+rs.getString(3)+"\t"+rs.getFloat(4));
            }
            if(count==0) System.out.println("No records found !");
        }
        void close() throws SQLException    {
            rs.close();
            stmt.close();
            con.close();
            sc.close();
        }
}
public class BankDatabase {
        public static void main(String args[]) throws SQLException   {
            boolean control=true;
            Bank bk=new Bank();
            System.out.println("Current records in database : ");
            bk.displayData();
            Scanner sc=new Scanner(System.in);
            while(control)  {
                try {
                        System.out.println("1.Insert   2.Delete   3.Display   Any other choice : Exit");
                        System.out.print("Enter your choice : ");
                        switch(sc.nextInt())    {
                                case 1 : bk.insertRow();
                                         break;
                                case 2 : bk.deleteRow();
                                         break;
                                case 3 : bk.displayData();
                                         break;
                                default: control=false;
                        }
                }
                catch(SQLException e)  {
                        System.out.println("SQLException : "+e);
                }
            }
            sc.close();
            bk.close();
        }
}
```

Output:



```
Current records in database :
No records found !
1.Insert   2.Delete   3.Display   Any other choice : Exit
Enter your choice : 1

Enter Account Number : 1
Enter First name : Shawn
Enter Surname : Miranda
Enter balance : 12000
1 row affected.

1.Insert   2.Delete   3.Display   Any other choice : Exit
Enter your choice : 1

Enter Account Number : 2
Enter First name : Ranjith
Enter Surname : Suvarna
Enter balance : 3000
1 row affected.
```



```
1.Insert   2.Delete   3.Display   Any other choice : Exit
Enter your choice : 3
1       Shawn   Miranda 12000.0
2       Ranjith Suvarna 3000.0
1.Insert   2.Delete   3.Display   Any other choice : Exit
Enter your choice : 2

Enter Account Number : 2
1 row deleted.

1.Insert   2.Delete   3.Display   Any other choice : Exit
Enter your choice : 3
1       Shawn   Miranda 12000.0
1.Insert   2.Delete   3.Display   Any other choice : Exit
Enter your choice : 4
```

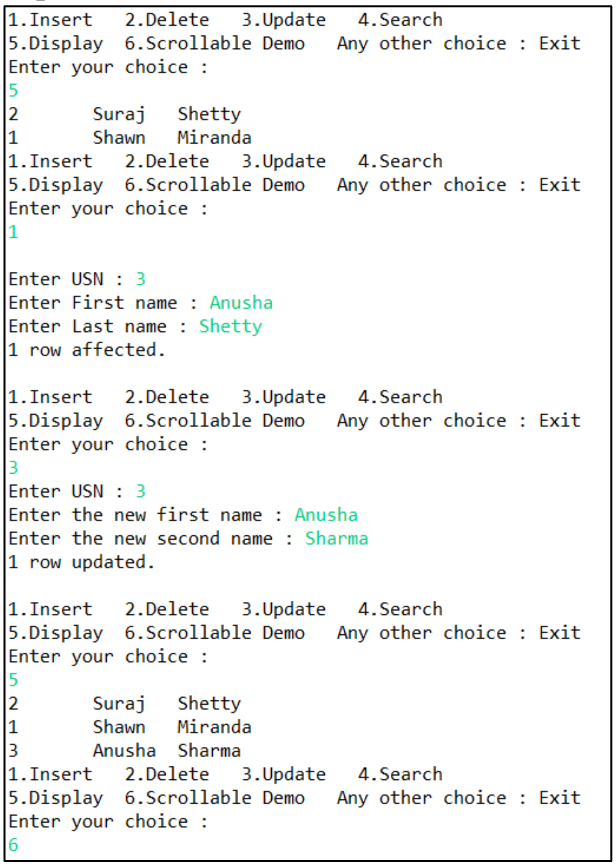
4. JDBC – Student Details

Write a JDBC Program to perform the following operation on database table student (usn,name).
1.Insert record 2.Delete 3.Update record 4.Search 5.Display 6.Demonstrate scrollable method

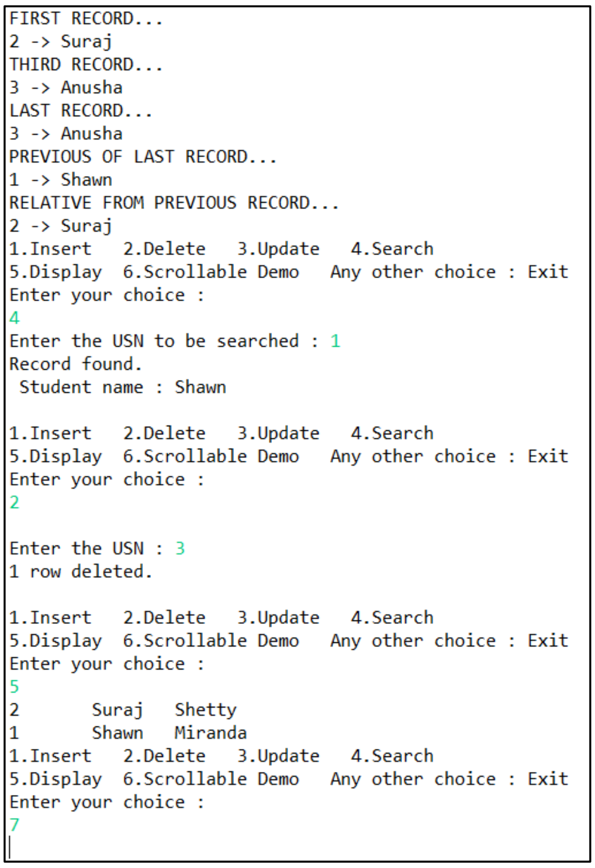
Program:

```
import java.sql.*;
import java.util.Scanner;
class Student  {
        private static Connection con;
        private static Statement stmt;
        private static ResultSet rs;
        private static Scanner sc;
        Student() throws SQLException   {
            try {
                Class.forName("com.mysql.jdbc.Driver ");
            }
            catch(ClassNotFoundException e)   {
                    System.out.println("Unable to load the driver!!");
                    System.exit(1);
            }
            try {
                    con=DriverManager.getConnection("jdbc:mysql://172.16.2.3/student ","student","student");
            }
            catch(SQLException e)   {
                    System.out.println("Cannot connect to the database !");
                    System.exit(1);
            }
            stmt=con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_UPDATABLE);
        }

        void insertRow() throws SQLException   {
                sc=new Scanner(System.in);
                System.out.print("\nEnter USN : ");
                int usn=sc.nextInt();
                sc.nextLine();
                System.out.print("Enter First name : ");
                String fname=sc.nextLine();
                System.out.print("Enter Last name : ");
                String lname=sc.nextLine();
                String insert="INSERT INTO Student VALUES("+usn+",'"+fname+"','"+lname+"');";
                int i=stmt.executeUpdate(insert);
                if(i>0) System.out.println("1 row affected.\n");
        }

        void deleteRow() throws SQLException   {
                sc=new Scanner(System.in);
                System.out.print("\nEnter the USN : ");
                int usn=sc.nextInt();
                String delete="DELETE FROM Student WHERE usn="+usn;
                int i=stmt.executeUpdate(delete);
                if(i>0) System.out.println("1 row deleted.\n");
        }
        void updateRow() throws SQLException   {
                sc=new Scanner(System.in);
                System.out.print("Enter USN : ");
                int usn=sc.nextInt();
                System.out.print("Enter the new first name : ");
                sc.nextLine();
                String fname=sc.nextLine();
                System.out.print("Enter the new second name : ");
                String lname=sc.nextLine();
                String update="UPDATE Student SET fname='"+fname+"', lname='"+lname+"' WHERE usn="+usn;
                int i=stmt.executeUpdate(update);
                if(i>0)  System.out.println("1 row updated.\n");
        }
        void search() throws SQLException   {
                sc=new Scanner(System.in);
                System.out.print("Enter the USN to be searched : ");
                int usn=sc.nextInt();
                String search="SELECT * FROM Student WHERE usn="+usn;
                rs=stmt.executeQuery(search);
                if(rs.next())
                        System.out.println("Record found.\n Student name : "+rs.getString(2)+"\n");
                else
                        System.out.println("No record found!\n");
        }
        void displayData() throws SQLException   {
                String select="SELECT * FROM Student";
                rs=stmt.executeQuery(select);
                while(rs.next())   {
                        System.out.println(rs.getInt(1)+"\t"+rs.getString(2)+"\t"+rs.getString(3));
                }
        }
        void scrollDemo() throws SQLException  {
                rs.first();
                System.out.println("FIRST RECORD...\n"+rs.getInt(1)+" -> "+rs.getString(2));
                rs.absolute(3);
                System.out.println("THIRD RECORD...\n"+rs.getInt(1)+" -> "+rs.getString(2));
                rs.last();
                System.out.println("LAST RECORD...\n"+rs.getInt(1)+" -> "+rs.getString(2));
                rs.previous();
                System.out.println("PREVIOUS OF LAST RECORD...\n"+rs.getInt(1)+" -> "+rs.getString(2));
                rs.relative(-1);
                System.out.println("RELATIVE FROM PREVIOUS RECORD...\n"+rs.getInt(1)+" -> "+rs.getString(2));
        }
        void close() throws SQLException  {
                if(rs!=null) rs.close();   //Comparision is to avoid closing of rs without initialising it
                stmt.close();
                con.close();
        }
}
public class StudentDatabase {
        public static void main(String args[]) throws SQLException  {
                boolean control=true;
                Student st=new Student();
                Scanner sc=new Scanner(System.in);
        while(control)  {
                try {
                        System.out.println("1.Insert  2.Delete  3.Update  4.Search\n  5.Display  6.Scrollable
                                                                Demo  Any other choice : Exit");
                        System.out.println("Enter your choice : ");
                        switch(sc.nextInt())  {
                                case 1:st.insertRow();
                                            break;
                                case 2:st.deleteRow();
                                            break;
                                case 3:st.updateRow();
                                            break;
                                case 4:st.search();
                                            break;
                                case 5:st.displayData();
                                            break;
                                case 6:st.scrollDemo();
                                            break;
                                default: control=false;
                        }
                }
                catch(SQLException e)  {
                        System.out.println("SQLException : "+e);
                }
        }
        sc.close();
        st.close();
    }
}
```

Output:



```
1.Insert   2.Delete   3.Update   4.Search
5.Display  6.Scrollable Demo   Any other choice : Exit
Enter your choice :
5
2       Suraj   Shetty
1       Shawn   Miranda
1.Insert   2.Delete   3.Update   4.Search
5.Display  6.Scrollable Demo   Any other choice : Exit
Enter your choice :
1

Enter USN : 3
Enter First name : Anusha
Enter Last name : Shetty
1 row affected.

1.Insert   2.Delete   3.Update   4.Search
5.Display  6.Scrollable Demo   Any other choice : Exit
Enter your choice :
3
Enter USN : 3
Enter the new first name : Anusha
Enter the new second name : Sharma
1 row updated.

1.Insert   2.Delete   3.Update   4.Search
5.Display  6.Scrollable Demo   Any other choice : Exit
Enter your choice :
5
2       Suraj   Shetty
1       Shawn   Miranda
3       Anusha  Sharma
1.Insert   2.Delete   3.Update   4.Search
5.Display  6.Scrollable Demo   Any other choice : Exit
Enter your choice :
6
```



```
FIRST RECORD...
2 -> Suraj
THIRD RECORD...
3 -> Anusha
LAST RECORD...
3 -> Anusha
PREVIOUS OF LAST RECORD...
1 -> Shawn
RELATIVE FROM PREVIOUS RECORD...
2 -> Suraj
1.Insert   2.Delete   3.Update   4.Search
5.Display  6.Scrollable Demo   Any other choice : Exit
Enter your choice :
4
Enter the USN to be searched : 1
Record found.
 Student name : Shawn

1.Insert   2.Delete   3.Update   4.Search
5.Display  6.Scrollable Demo   Any other choice : Exit
Enter your choice :
2

Enter the USN : 3
1 row deleted.

1.Insert   2.Delete   3.Update   4.Search
5.Display  6.Scrollable Demo   Any other choice : Exit
Enter your choice :
5
2       Suraj   Shetty
1       Shawn   Miranda
1.Insert   2.Delete   3.Update   4.Search
5.Display  6.Scrollable Demo   Any other choice : Exit
Enter your choice :
7
```

5. TCP socket program

Create a TCP socket program to send and receive message between server and client such that server echo back the message sent by the client.

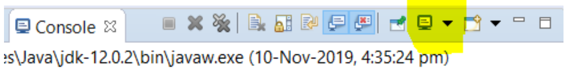
Note:
If you are unable to connect to the port please use different port in each run of program. The reason for this is, the client program terminates but server program will be in continuous executing unless you manually terminate. So it will be using that port number. If you want to use same port number, manually terminate program in eclipse console.

Program:

```
import java.io.*;      //Server Side program
import java.net.*;
import java.util.Scanner;
public class Server {
        private static ServerSocket serverSocket;
        private static final int PORT=1234;
        public static void main(String args[])
        {
                System.out.println("Opening port...");
                try {
                        serverSocket=new ServerSocket(PORT);
                }
                catch(IOException e)   {
                        System.out.println("Unable to connect attach port.");
                        System.exit(0);
                }
                do {
                        handleClient();
                } while(true);
        }
        private static void handleClient()
        {
                Socket link=null;
                try {
                        link=serverSocket.accept();
                        Scanner input=new Scanner(link.getInputStream());
                        PrintWriter output=new PrintWriter(link.getOutputStream(),true);
                        int messageCount=0;
                        String message=input.nextLine();
                        while(!message.equals("***CLOSE***"))   {
                                System.out.println("Message received");
                                messageCount++;
                                output.println("Message "+messageCount+" : "+message);
                                message=input.nextLine();
                        }
                        output.println(messageCount+" messages received.");
                }
                catch(IOException e)  {
                        e.printStackTrace();
                }
                finally {
                        try {
                                System.out.println("\nClosing connection..");
                                link.close();
                        }
                        catch(IOException e)  {
                                System.out.println("Unable to disconnect.");
                                System.exit(1);
                        }
                }
        }
}
```

```
import java.net.*;   //Client side program
import java.util.Scanner;
import java.io.*;
public class Client {
        private static final int PORT=1234;
        private static InetAddress host;
        public static void main(String args[])  {
                try {
                        host=InetAddress.getLocalHost();
                }
                catch(UnknownHostException e)  {
                        System.out.println("HOST id not found!");
                        System.exit(1);
                }
                accessServer();
        }
        public static void accessServer()  {
                Socket link=null;
                try {
                        link=new Socket(host,PORT);
                        Scanner input=new Scanner(link.getInputStream());
                        PrintWriter output=new PrintWriter(link.getOutputStream(),true);
                        Scanner userEntry=new Scanner(System.in);
                        String message,response;
                        do {
                                System.out.println("Enter message : ");
                                message=userEntry.nextLine();
                                output.println(message);
                                response=input.nextLine();
                                System.out.println("\nServer > "+response);
                        } while(!message.equals("***CLOSE***"));
                }
                catch(IOException e)  {
                        e.printStackTrace();
                }
                finally {
                        try {
                                System.out.println("\n*Closing the connection .. *");
                                link.close();
                        }
                        catch(IOException e)  {
                                System.out.println("Unable to disconnect.");
                                System.exit(1);
                        }
                }
        }
}
```

Note:
Port number of client program should be always same as the port number of server side program.
First run the server program, then run the client code.
If you are unable to see both outputs i.e server as well as client, switch the console to next page.



```
:s\Java\jdk-12.0.2\bin\javaw.exe (10-Nov-2019, 4:35:24 pm)
```

Output:

```
Opening port...
Message received
Message received

Closing connection..
```

```
Enter message :
Hello Server

Server > Message 1 : Hello Server
Enter message :
How are you?

Server > Message 2 : How are you?
Enter message :
***CLOSE***

Server > 2 messages received.

*Closing the connection .. *
```

6. UDP Socket Program

Create a TCP socket program to send and receive message between server and client such that server echo back the message sent by the client.

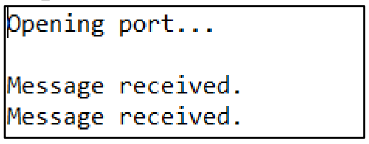
Note:
If you are unable to connect to the port please use different port in each run of program. The reason for this is, the client program terminates but server program will be in continuous executing unless you manually terminate. So it will be using that port number. If you want to use same port number, manually terminate program in eclipse console.

Program:

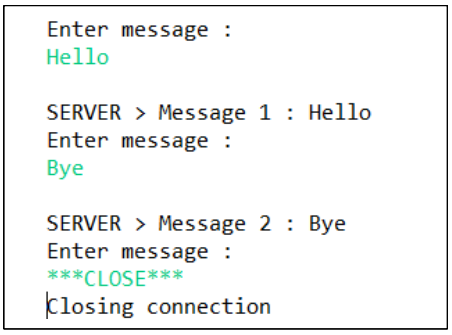
```
import java.io.*;  //Server side program
import java.net.*;

public class Server   {
        private static final int PORT=1234;
        private static DatagramSocket datagramSocket;
        private static DatagramPacket inPacket,outPacket;
        private static byte[] buffer;
        public static void main(String args[])  {
                System.out.println("Opening port...\n");
                try {
                        datagramSocket=new DatagramSocket(PORT);
                }
                catch(SocketException e)  {
                        System.out.println("Unable to attach to the port!");
                        System.exit(1);
                }
                handleClient();
        }

        private static void handleClient()  {
                try {
                        String messageIn,messageOut;
                        int messageCount=0;
                        do {
                                buffer=new byte[256];
                                inPacket=new DatagramPacket(buffer,buffer.length);
                                datagramSocket.receive(inPacket);
                                InetAddress clientAddress=inPacket.getAddress();
                                int clientPort=inPacket.getPort();
                                messageIn=new String(inPacket.getData(),0,inPacket.getLength());
                                System.out.println("Message received.");
                                messageCount++;
                                messageOut="Message "+messageCount+" : "+messageIn;
                                outPacket=new DatagramPacket(messageOut.getBytes(),messageOut.length(),
                                                                        clientAddress,clientPort);
                                datagramSocket.send(outPacket);
                        } while(true);
                }
                catch(IOException e)  {
                        e.printStackTrace();
                }
                finally {
                        System.out.println("\nClosing connection...");
                        datagramSocket.close();
                }
        }
}
```

```java
import java.io.*;
import java.net.*;
import java.util.Scanner;
public class Client {
    private static final int PORT=1234;
    private static InetAddress host;
    private static DatagramSocket datagramSocket;
    private static DatagramPacket inPacket,outPacket;
    private static byte[] buffer;
    public static void main(String args[])      {
        try {
            host=InetAddress.getLocalHost();
        }
        catch(UnknownHostException e)  {
            System.out.println("Host ID not found!");
            System.exit(1);
        }
        accessServer();
    }
    private static void accessServer()  {
        try {
            datagramSocket=new DatagramSocket();
            Scanner userEntry=new Scanner(System.in);
            String message,response;
            do {
                System.out.println("Enter message : ");
                message=userEntry.nextLine();
                if(!message.equals("***CLOSE***"))  {
                    outPacket=new DatagramPacket(message.getBytes(),message.length(),
                                                                        host,PORT);
                    datagramSocket.send(outPacket);
                    buffer=new byte[256];
                    inPacket=new DatagramPacket(buffer,buffer.length);
                    datagramSocket.receive(inPacket);
                    response=new String(inPacket.getData(),0,inPacket.getLength());
                    System.out.println("\nSERVER > "+response);
                }
            } while(!message.equals("***CLOSE***"));
        }
        catch(IOException e)  {
            e.printStackTrace();
        }
        finally {
            System.out.println("Closing connection");
            datagramSocket.close();
        }
    }
}
```

Output:



```
Opening port...

Message received.
Message received.
```



```
Enter message :
Hello

SERVER > Message 1 : Hello
Enter message :
Bye

SERVER > Message 2 : Bye
Enter message :
***CLOSE***
Closing connection
```

7. Servlet Login

Build a Java Program for Login with password validation using servlets.

Program:

Login.html

```
<html>
        <head>
                <title>Login</title>
        </head>

        <body>
                <form method="post" action="LoginHandler">
                        Name:<input type="text" name="name">
                        Password:<input type="password" name="password">
                        <input type="submit" value="LOGIN">
                </form>
        </body>
</html>
```

LoginHandler.java

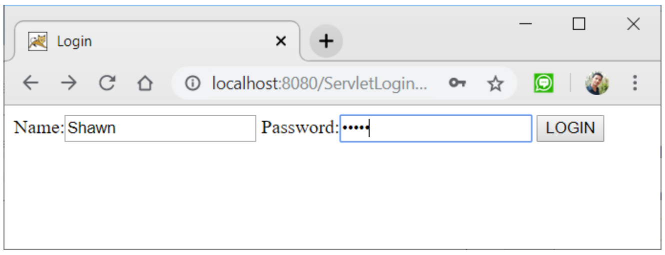
```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class LoginHandler extends HttpServlet {
        public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException
         {
                res.setContentType("text/html");
                PrintWriter out=res.getWriter();
                String uname=req.getParameter("name");
                String password=req.getParameter("password");
                String name="NMAMIT";
                String pass="1234";
                out.println("<HTML>");
                out.println("<body>");
                if(uname.equals(name) && password.equals(pass))
                        out.println("<h1>Hello!! Welcome ot NMAMIT.</h1>");
                else    {
                        out.println("<h1>Your username and password are invalid.</h1>");
                        out.println("You may want to <a href=\"login.html\">Try again</a>");
                }
                out.println("</body>");
                out.println("</html>");
                out.close();
        }
}
```

web.xml

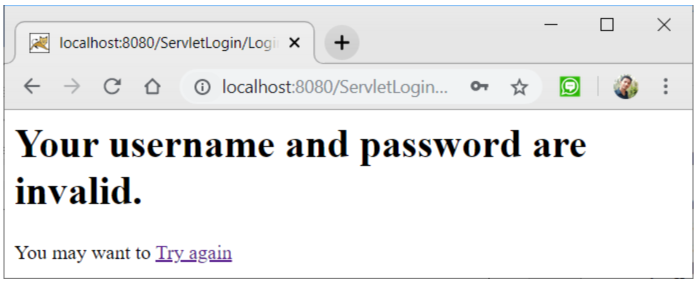
```
<servlet>
        <servlet-name>LoginHandler</servlet-name>
        <servlet-class>LoginHandler</servlet-class>
  </servlet>
```

```
<servlet-mapping>
    <servlet-name>LoginHandler</servlet-name>
    <url-pattern>/LoginHandler</url-pattern>
</servlet-mapping>
```

Output:

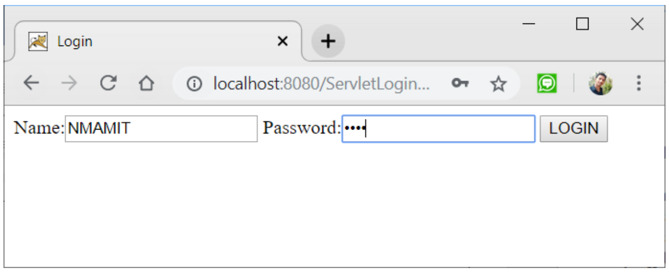


Name: Shawn Password: ••••• LOGIN

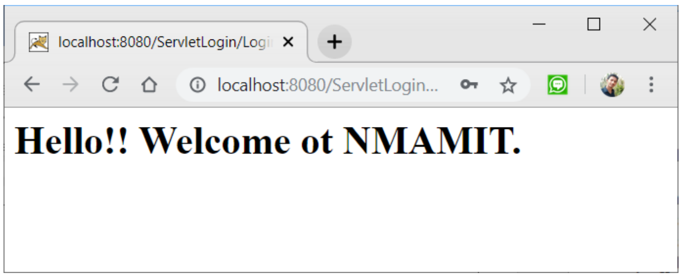

localhost:8080/ServletLogin/Logi
localhost:8080/ServletLogin...

Your username and password are invalid.

You may want to Try again



Name: NMAMIT Password: •••• LOGIN


localhost:8080/ServletLogin/Logi
localhost:8080/ServletLogin...

Hello!! Welcome ot NMAMIT.

8. Arithmetic Operations using JSP

Design a form using HTML which contains two textboxes and a submit button to read two numbers. Perform Addition, Subtraction, Multiplication or Division base on users choice through radio buttons. Display result using GET or PORT methods.

Program:

arithmetic.html

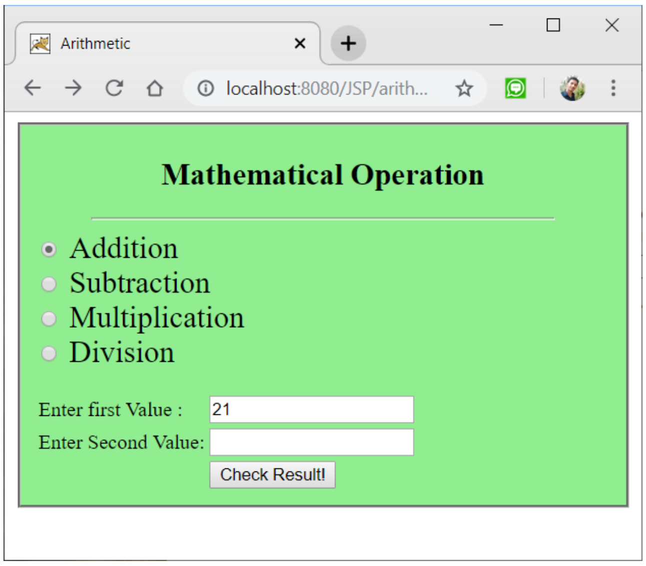
```
<html>
    <title>Arithmetic</title>
    <body>
        <form method="post" action="index.jsp">
            <fieldset style="width:30%; background-color:lightgreen ">
                    <h2><center>Mathematical Operation</center></h2>
                    <hr style="width:80%">
                    <font size=5 face="Times New Roman">
                        <input type="radio" name="selected" value="Add" checked> Addition</input><br>
                        <input type="radio" name="selected" value="Sub"> Subtraction</input><br>
                        <input type="radio" name="selected" value="Mul"> Multiplication</input><br>
                        <input type="radio" name="selected" value="Div"> Division </input><br>
                    </font>
                    <table>
                        <tr>
                            <td>Enter first Value :</td>
                            <td><input type="text" name="op1"></td>
                        </tr>
                        <tr>
                            <td>Enter Second Value:</td>
                            <td><input type="text" name="op2"> </td>
                        </tr>
                        <br>
                        <tr>
                            <td></td>
```

```
                                        <td><input type="submit" name="result" value="Check Result!"></td>
                                </tr>
                        </table>
                </fieldset>
            </form>
        </body>
</html>
```

index.jsp

```
<html>
        <head>
                <title>Result</title>
        <head>
        <body>
                <h1><center>Result for <%=request.getParameter("selected")%></center></h1>
                <%
                 String num1=request.getParameter("op1");
                 String num2=request.getParameter("op2");
                 int i=Integer.parseInt(num1);
                 int j=Integer.parseInt(num2);
                 int k=0;
                 String str=request.getParameter("selected");
                 if(str.equals("Add"))
                        k=i+j;
                 else if(str.equals("Sub"))
                        k=i-j;
                 else if(str.equals("Mul"))
                        k=i*j;
                 else if(str.equals("Div"))
                        k=i/j;
                %>
                Result is : <%=k%>
        </body>
</html>
```

Output:

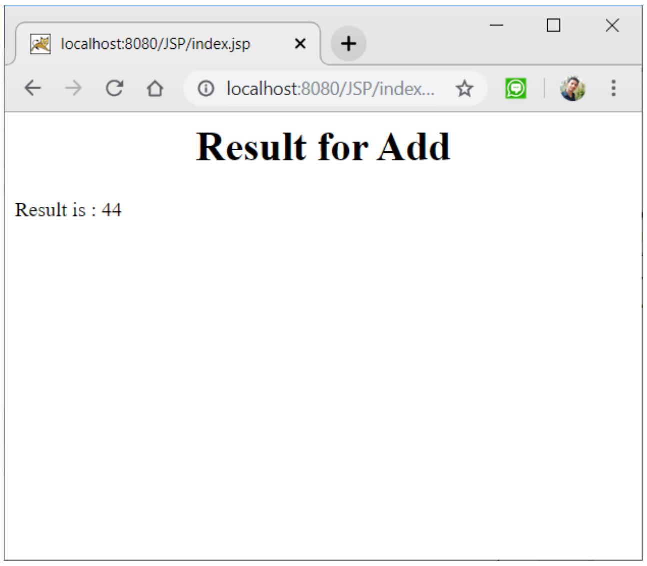


Mathematical Operation

- [✓ Addition]
- Subtraction
- Multiplication
- Division

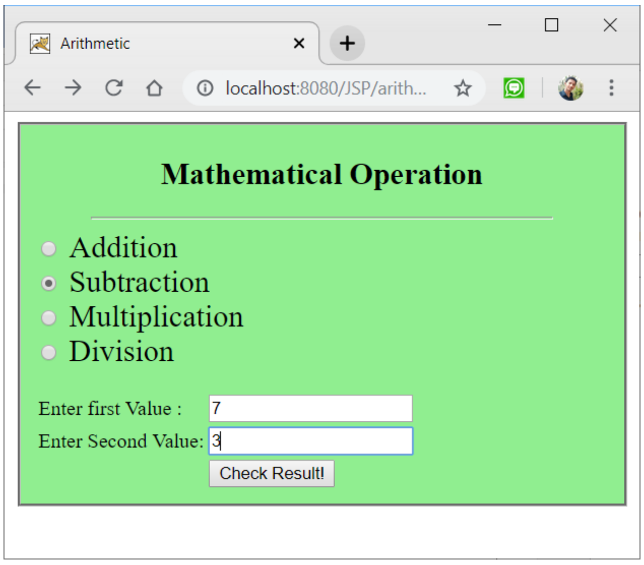
Enter first Value : 21
Enter Second Value:

Check Result!



Result for Add

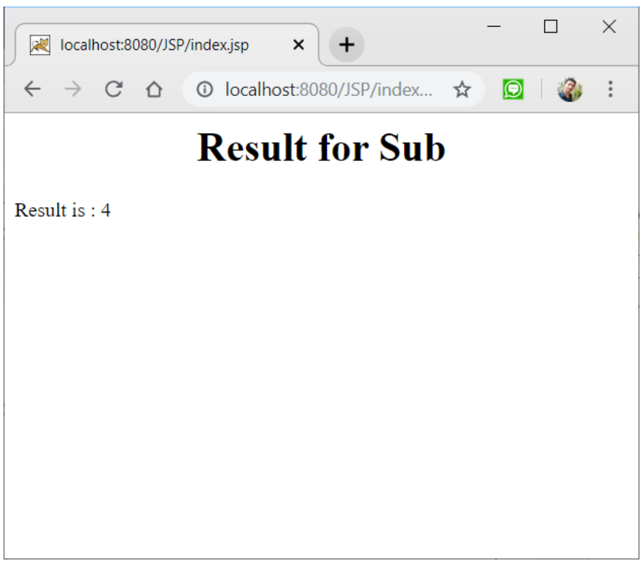
Result is : 44



Mathematical Operation

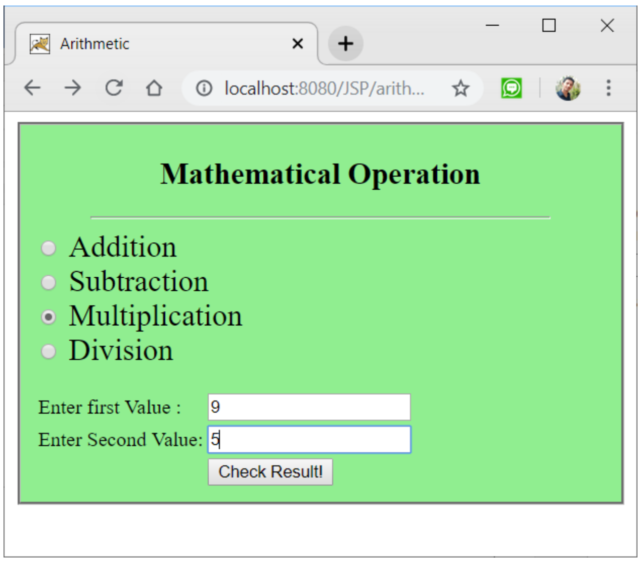
- ( ) Addition
- [✓ Subtraction]
- ( ) Multiplication
- ( ) Division

Enter first Value :	7
Enter Second Value:	3



Result for Sub

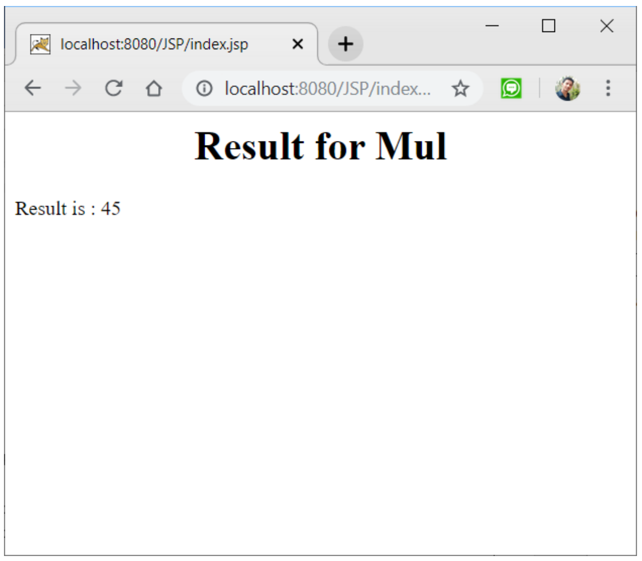
Result is : 4



Mathematical Operation

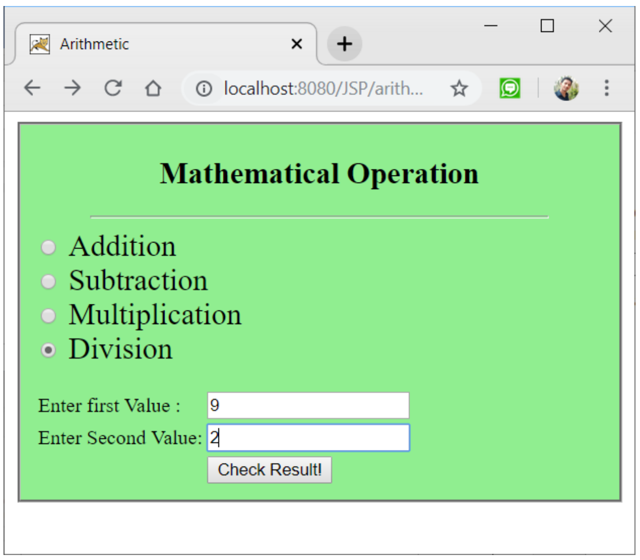
- ( ) Addition
- ( ) Subtraction
- [✓ Multiplication]
- ( ) Division

Enter first Value :	9
Enter Second Value:	5



Result for Mul

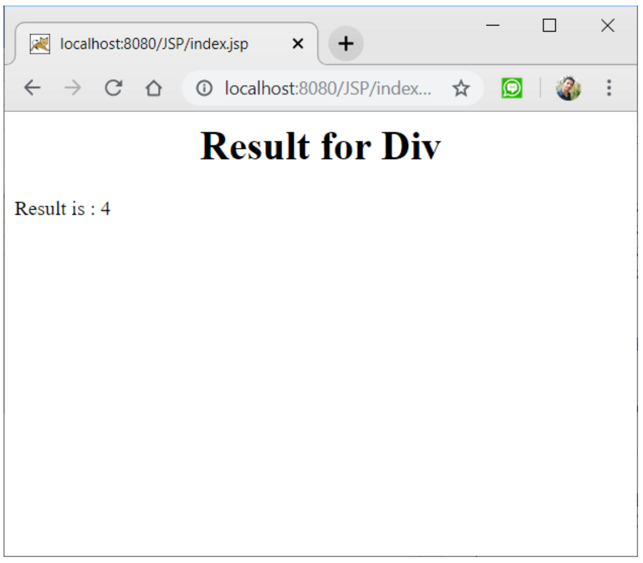
Result is : 45



Mathematical Operation

- ( ) Addition
- ( ) Subtraction
- ( ) Multiplication
- [✓ Division]

Enter first Value :	9
Enter Second Value:	2



Result for Div

Result is : 4

***