LAB 1

TITLE: Create a custom Exception class. You need to consider two integer inputs which must be supplied by the user. You will display the sum of the integers if and only if the sum is less than 100. If it is not less than 100, throw your custom exception.

CODE:

SumException

```
package Exception;

import java.util.Scanner;

public class SumException extends Exception {

   String msg;

   SumException(String msg){

      this.msg=msg;

 }

}

interface Demo

{

   int sum(int x, int y) throws SumException;

}

class SUM implements Demo

{

   @Override

   public int sum(int x, int y) throws SumException {

      int sumofIntegers=x+y;

      if(sumofIntegers<=100){

         return sumofIntegers;
```

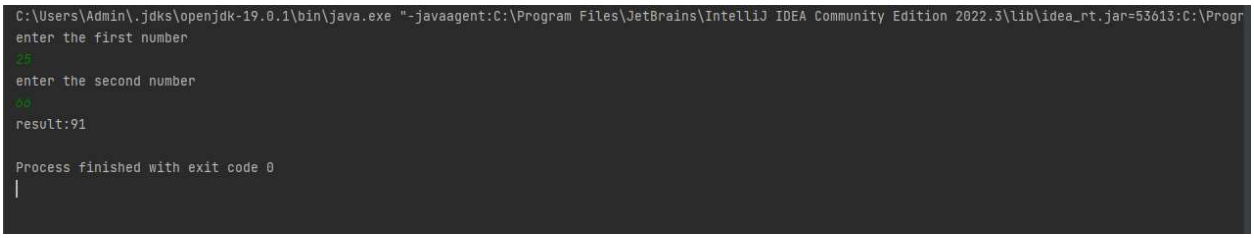
```
        }
        Else
           {
         throw new SumException("sum is greater than 100");
        }
    }
}
```

CustomException

```
package Exception;
import java.util.Scanner;
public class CustomException {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int number1, number2;
        System.out.println("enter the first number");
        number1 = Integer.parseInt(in.nextLine());
        System.out.println("enter the second number");
        number2 = Integer.parseInt(in.nextLine());
        SUM add = new SUM();
        try {
            int result = add.sum(number1, number2);
            System.out.println("result:" + result);
        } catch (SumException e) {
            System.out.println("Caught the custom exception: " + e);
```

```
            e.printStackTrace();
        }
    }
}
```

OUTPUT



LAB 2

TITLE: Write a program to print "Good morning" and "Welcome" continuously on the screen in Java using threads

CODE:

Thread

```
package Thread;

public class thread extends Thread{

   @Override

   public void run() {

      for(int i=0; i<100; i++){

         try{

            thread.sleep(1000);

         }

         catch(InterruptedException e){

            System.out.println(e);

         }

         System.out.println("Good morning");

      }

   }

}
```

```
package Thread;

public class Two extends Thread {

   public void run() {

      for (int i = 0; i < 100; i++) {

         try {
```

```
            thread.sleep(1000);
        } catch (InterruptedException e) {
            System.out.println(e);
        }
        System.out.println("Welcome");
    }
  }
}
```

Main

```
package Thread;
public class main {
   public static void main(String[] args) {
      thread obj = new thread();
      Two obj1 = new Two();
      Thread th= new Thread();
      obj.setName("thread");
      obj1.setName("Two");
      System.out.println(obj);
      System.out.println(obj1);
      obj.start();
      obj1.start();
   }
}
```

OUTPUT

```
C:\Users\Admin\.jdks\openjdk-19.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.3\lib\idea_rt.jar=64043:C:\Pro
Thread[#22,thread,5,main]
Thread[#23,Two,5,main]
Good morning
Welcome
Good morning
Welcome
Good morning
Welcome
Good morning
Welcome
Good morning
Welcome
Good morning
Welcome
Good morning
Welcome
Good morning
Welcome
Good morning
Welcome
Good morning
Welcome
Good morning
Welcome
Good morning
Welcome
```

LAB 3

TITLE: Demonstrate gerPriority () and setPriority () methods in Java threads

CODE:

```
package Thread;

public class Priority extends Thread{

    public void run(){

        System.out.println("running....");

    }

    public static void main(String[] args) {

        Priority obj= new Priority();

        Priority obj1 = new Priority();

        obj.setPriority(5);

        obj1.setPriority(8);

        System.out.println("priority of thread obj: "+obj.getPriority());

        System.out.println("prioriy of thread obj2: "+obj1.getPriority());

        obj.start()

 }

}
```

OUPUT

```
C:\Users\Admin\.jdks\openjdk-19.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.3\lib\idea_rt.jar=53601:C:\Prc
priority of thread obj: 8
prioriy of thread obj2: 10
running....

Process finished with exit code 0
```

LAB 4

TITLE: Demonstrate Thread Synchronization in java

CODE

Class caller:

```
package Multithreading.Synchronization;
public class Caller implements Runnable{
    String message;
    CallMe target;
    Thread t;
    public Caller(CallMe target, String message) {
        this.target = target;
        this.message = message;
        t = new Thread(this);
        t.start();
    }
    public void run() {
       target.call(this.message);
    }
}
```

Class caller:

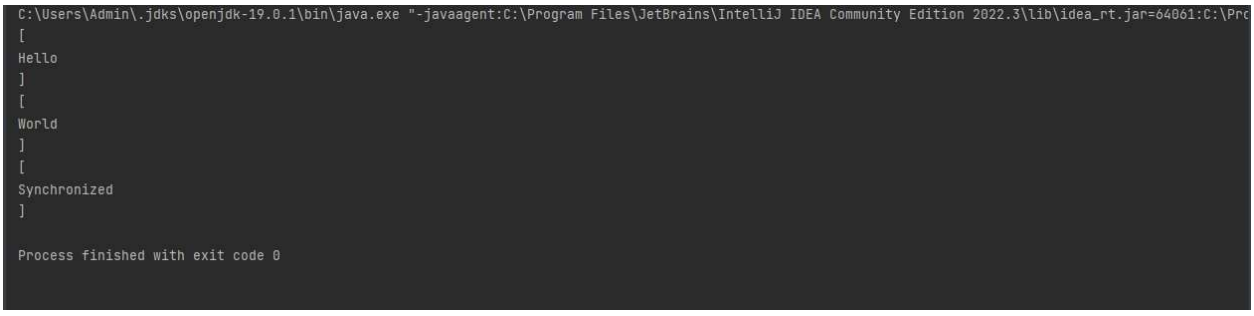
```
package Multithreading.Synchronization;
public class CallMe {
   synchronized void call(String msg) {
        System.out.println("[" );
       System.out.println(msg);
        try {
```

```
            Thread.sleep(5000)   }
           catch(InterruptedException e) {
            System.out.println("Interrupted");
        }
        System.out.println("]");
    }
}
```

Class main:

```
package Multithreading.Synchronization;
public class Main {
    public static void main(String[] args) {
        CallMe target = new CallMe();
        new Caller(target, "Hello");
        new Caller(target, "Synchronized");
        new Caller(target, "World");
    }
    }
```

Output:



LAB 5

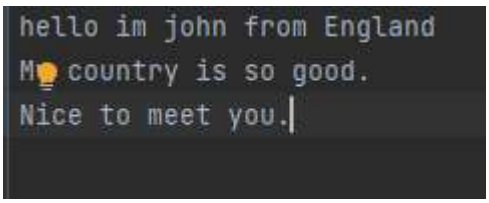
TITLE: Randomly access a file named test.txt and write its contents from 20th position to a new file called output.txt

CODE:

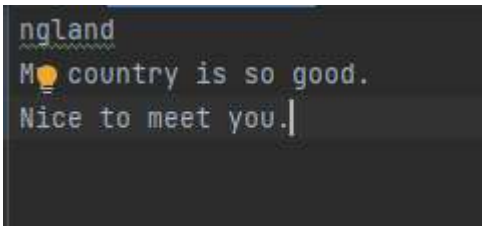
```
package FileHandling.RandomAccess;

import java.io.File;

import java.io.IOException;

import java.io.RandomAccessFile;

public class Main {

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

      File file = new File("Test.txt");

      File file1 = new File("output.txt");

      RandomAccessFile randomAccessFile = new
RandomAccessFile(file,"r");

      RandomAccessFile randomAccessFile1 = new
RandomAccessFile(file1,"rw");

      randomAccessFile.seek(20);

      int temp;

      while ((temp=randomAccessFile.read())!=-1)

      {

         randomAccessFile1.write(temp);

      }

      randomAccessFile.close();

      randomAccessFile1.close();

   }

}
```

Output

Test.txt



Output.txt



LAB 6

TITLE: Create a class Employee that has its members name and age. Create an object of employee class and write it into a file named employee.txt. After writing read back that file and print the name and age of that employee object

CODE:

Employee

```
package FileHandling.WritingAndReadingObjects;

import java.io.Serializable;

public class Employee implements Serializable {

    private Integer age;

    private String name;

    public Integer getAge() {

        return age;

    }

   public void setAge(Integer age) {

        this.age = age;

    }

    public String getName() {

        return name;

    }

    public void setName(String name) {

 this.name = name;

    }

}
```

Reading

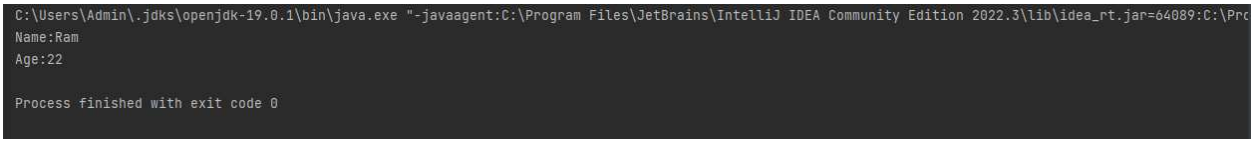
```
package FileHandling.WritingAndReadingObjects;

import java.io.FileInputStream;

import java.io.IOException;

import java.io.ObjectInputStream;

public class Reading {

   public static void main(String[] args) throws
IOException,ClassNotFoundException {

      FileInputStream fileInputStream = new

            FileInputStream("employee.txt");

      ObjectInputStream objectInputStream =

            new ObjectInputStream(fileInputStream);

      Employee employee = (Employee) objectInputStream.readObject();

      System.out.println("Name:" +employee.getName());

      System.out.println("Age:"+employee.getAge());

      objectInputStream.close();

      fileInputStream.close();

   }

}
```

Writing:

```
package FileHandling.WritingAndReadingObjects;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.ObjectOutputStream;

public class Writing {
```

```
    public static void main(String[] args) throws IOException {
        Employee employee = new Employee();
        employee.setAge(22);
        employee.setName("Ram");
        FileOutputStream fileOutputStream = new
FileOutputStream("employee.txt");
        ObjectOutputStream objectOutputStream = new
ObjectOutputStream(fileOutputStream);
        objectOutputStream.writeObject(employee);
        System.out.println("Done");
        objectOutputStream.close();
        fileOutputStream.close();


    }
}
```

Output:



JAVA SWING

LAB 7

TITLE: Write a GUI program using components to find sum and difference of two numbers. Use two text fields for giving input and a label for output. The program should display sum if user presses mouse and difference if user release mouse.

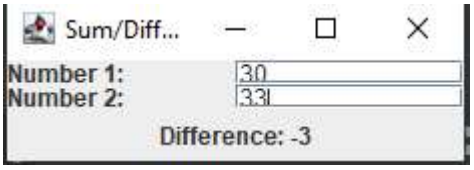
Code:

```
package EventHandling.Examples;

import javax.swing.*;

import java.awt.*;

import java.awt.event.*;

public class SumDifference extends JFrame implements
MouseListener {

   private JTextField num1Field, num2Field;

   private JLabel resultLabel;

   public SumDifference() {

      super("Sum/Difference Calculator");

      JLabel label1 = new JLabel("Enter first number:");

      JLabel label2 = new JLabel("Enter second number:");

      num1Field = new JTextField(10);

      num2Field = new JTextField(10);

      JButton button = new JButton("Calculate");
```

```
    JPanel panel = new JPanel(new GridLayout(3, 2));
    panel.add(label1);
    panel.add(num1Field);
    panel.add(label2);
    panel.add(num2Field);
    panel.add(button);
    resultLabel = new JLabel();
    Container contentPane = getContentPane();
    contentPane.setLayout(new BorderLayout());
    contentPane.add(panel, BorderLayout.NORTH);
    contentPane.add(resultLabel, BorderLayout.CENTER);
    button.addMouseListener(this);
    setSize(300, 150);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setVisible(true);
}
public void mouseClicked(MouseEvent e) {
    try {
        double num1 = Double.parseDouble(num1Field.getText());
        double num2 = Double.parseDouble(num2Field.getText());
        if (SwingUtilities.isLeftMouseButton(e)) {
```

```
            double result = num1 + num2;
            resultLabel.setText("Sum: " + result);
        } else {
          double result = num1 - num2;
            resultLabel.setText("Difference: " + result);
        }
    } catch (NumberFormatException ex) {
        resultLabel.setText("Invalid input");
    }
  }
  public void mouseEntered(MouseEvent e) {}
  public void mouseExited(MouseEvent e) {}
  public void mousePressed(MouseEvent e) {}
  public void mouseReleased(MouseEvent e) {}
  public static void main(String[] args) {
    new SumDiffCalculator();
  }
```

Output



LAB 8

TITLE: create a table named Movie (id, Tille, Genre, Language, Length). Write a program to design a GUI form to take input for this table and insert the data into table after clicking the OK button

CODE:

```
package EventHandling.Examples;

import javax.swing.*;

import java.awt.*;

import java.awt.event.*;

import java.sql.*;

public class MovieForm extends JFrame implements ActionListener {

   private JTextField idField, titleField, genreField, languageField, lengthField;

   private JLabel idLabel, titleLabel, genreLabel, languageLabel, lengthLabel;

   private JButton okButton;

   public MovieForm() {

      super("Add Movie");

      idLabel = new JLabel("ID:");

      idField = new JTextField(10);

      titleLabel = new JLabel("Title:");

      titleField = new JTextField(10);
```

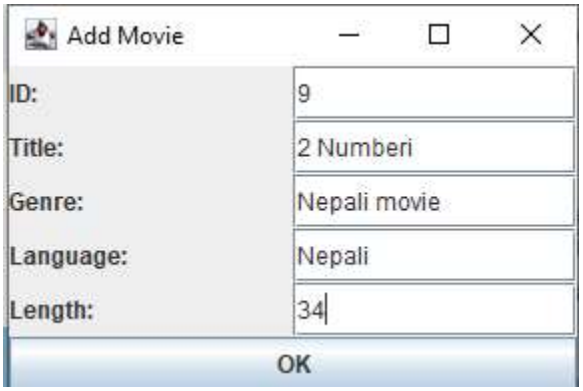
```
genreLabel = new JLabel("Genre:");

genreField = new JTextField(10);

languageLabel = new JLabel("Language:");

languageField = new JTextField(10);

lengthLabel = new JLabel("Length:");

lengthField = new JTextField(10);

okButton = new JButton("OK");

JPanel panel = new JPanel(new GridLayout(5, 2));

panel.add(idLabel);

panel.add(idField);

panel.add(titleLabel);

panel.add(titleField);

panel.add(genreLabel);

panel.add(genreField);

panel.add(languageLabel);

panel.add(languageField);

panel.add(lengthLabel);

panel.add(lengthField);

Container contentPane = getContentPane();

contentPane.setLayout(new BorderLayout());

contentPane.add(panel, BorderLayout.CENTER);
```

```
        contentPane.add(okButton, BorderLayout.SOUTH);
        okButton.addActionListener(this);
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == okButton) {
            try {
                int id = Integer.parseInt(idField.getText());
                String title = titleField.getText();
                String genre = genreField.getText();
                String language = languageField.getText();
                int length = Integer.parseInt(lengthField.getText());
                String url = "jdbc:mysql://localhost:3306/mydatabase";
                String username = "root";
                String password = "";
                Connection conn = DriverManager.getConnection(url,
username, password);
                Statement stmt = conn.createStatement();
```

```
            String query = String.format("INSERT INTO Movie (id, Title,
        Genre, Language,     Length) VALUES (%d, '%s', '%s', '%s', %d)",
        id, title, genre, language, length);

            stmt.executeUpdate(query);

            JOptionPane.showMessageDialog(this, "Movie added to
database.");

            conn.close();

        } catch (NumberFormatException ex) {

            JOptionPane.showMessageDialog(this, "Invalid input.");

        } catch (SQLException ex) {

            JOptionPane.showMessageDialog(this, "Database error: " +
ex.getMessage());

        }

    }

}

public static void main(String[] args) {

    new MovieForm();

}

}
```

OUTPUT:

In Frame



In database:

	id	Title	Genre	Language	Length
Edit Copy Delete	1	Movie	Movie	Test	12
Edit Copy Delete	2	iron man	marvel studio	english	30
Edit Copy Delete	3	jari	kabaddi series	nepali	34
Edit Copy Delete	6	bulaki	nepali movie	Nepali	34
Edit Copy Delete	8	bachelor	Nepali	Nepali	33
Edit Copy Delete	9	2 Numberi	Nepali movie	Nepali	34

LAB 9

TITLE: Fetch above Movie records from database and display the records in a JTable.

CODE:

```
package EventHandling.Examples;

import javax.swing.*;

import javax.swing.table.DefaultTableModel;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import java.sql.*;

public class Retrive {

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

       String[] columnNames = {"id", "Title", "Genre", "Language",
"Length"};

       JFrame jFrame = new JFrame("Movie");

       String url = "jdbc:mysql://localhost:3306/mydatabase";

       String username = "root";

       String password = "";

      Connection conn = DriverManager.getConnection(url, username,
password);
```

```
        Statement stmt = conn.createStatement();
        String sql = "select * from Movie";
        ResultSet resultSet = stmt.executeQuery(sql);
        JTable jTable = new JTable();
        DefaultTableModel model = new DefaultTableModel();
        model.setColumnIdentifiers(columnNames);
        jTable.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
        jTable.setModel(model);
        while(resultSet.next())
        {
            Integer id = resultSet.getInt("id");
            String Title = resultSet.getString("Title");
            String Genre = resultSet.getString("Genre");
            String Language = resultSet.getString("Language");
            Integer Length = resultSet.getInt("Length");
            model.addRow(new Object[]{id, Title, Genre, Language,
Length});
        }
//        making table scrollable
        JScrollPane jScrollPane = new JScrollPane(jTable);
        jFrame.add(jScrollPane);
```

```
        jFrame.setVisible(true);

        jFrame.setSize(400,400);

jFrame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);

    }

}
```

OUPUT:

id	Title	Genre	Language	Length
1	Movie	Movie	Test	12
2	iron man	marvel stu...	english	30
3	jari	kabaddi se...	nepali	34
6	bulaki	nepali movie	Nepali	34
8	bachelor	Nepali	Nepali	33
9	2 Numberi	Nepali movie	Nepali	34
98	ffff	ffff	ffff	666

Networking:
LAB 10

TITLE:URL“http://example.com:80/docs/books/tutorial/index.html?name=networking#DOWNLOADING” Display its protocol,authority,host,port,path,query string,file name and reference using URL class.

CODE:

```
package URL;

import java.net.MalformedURLException;

import java.net.URL;

public class URLClass {

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

        URL sampleUrl = new URL("http://example.com:80/docs" +

          "/books/tutorial/index.html?name=networking#DOWNLOADING");

        System.out.println("protocol = " + sampleUrl.getProtocol());

        System.out.println("authority = " + sampleUrl.getAuthority());

        System.out.println("host = " + sampleUrl.getHost());

        System.out.println("port = " + sampleUrl.getPort());

        System.out.println("path = " + sampleUrl.getPath());

        System.out.println("query = " + sampleUrl.getQuery());

        System.out.println("filename = " + sampleUrl.getFile());

        System.out.println("ref = " + sampleUrl.getRef());
```

```
    }
}
```

Output:

```
C:\Users\Admin\.jdks\openjdk-19.0.1\bin\java.exe "-javaagent:C:\Program
protocol = http
authority = example.com:80
host = example.com
port = 80
path = /docs/books/tutorial/index.html
query = name=networking
filename = /docs/books/tutorial/index.html?name=networking
ref = DOWNLOADING

Process finished with exit code 0
```

LAB 11:

TITLE: Write a simple program to chat between client and server using TCP socket.

Client

```
package TCPSocket.ChatClientServer;

import java.io.*;

import java.net.Socket;

import java.util.Scanner;

public class Client {

   private static  int port=8000;

   private static  String serverName="localhost";

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

      System.out.println("I am Client connecting to port: "

            +port+" server::"+serverName);

      Socket clientSocket = new Socket(serverName,port);

//    initialize input stream and output stream

      OutputStream outputStream = clientSocket.getOutputStream();

      DataOutputStream dataOutputStream = new
DataOutputStream(outputStream);

      InputStream inputStream = clientSocket.getInputStream();

      DataInputStream dataInputStream = new
DataInputStream(inputStream);
```

```
        Scanner scanner = new Scanner(System.in);
        String line = "";
        while (!line.equals("over"))
        {
//            write to server
            System.out.println("Enter Your Message :");
            line=scanner.nextLine();
            dataOutputStream.writeUTF(line);
//            read from server
            line = dataInputStream.readUTF();
            System.out.println("Message from server:: "+line);
        }
        dataOutputStream.close();
        clientSocket.close();
    }
}
```

Server

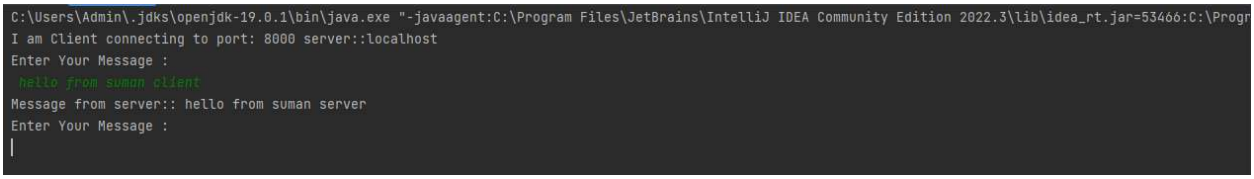
```
package TCPSocket.ChatClientServer;

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

```
import java.util.Scanner;


public class Server {
    public static void main(String[] args) throws IOException {
        ServerSocket serverSocket = new ServerSocket(8000);
        System.out.println("Server Started");
        System.out.println("Waiting for client");
        Socket socket = serverSocket.accept();
//        initialize Input Stream And Output Stream
        InputStream inputStream = socket.getInputStream();
        DataInputStream dataInputStream = new
DataInputStream(inputStream);
        OutputStream outputStream = socket.getOutputStream();
        DataOutputStream dataOutputStream = new
DataOutputStream(outputStream);
        Scanner scanner = new Scanner(System.in);
        String line = "";
        while(!line.equals("over")){
//            receive from client
            line = dataInputStream.readUTF();
            System.out.println("Message From Client::: "+line);
//            write to client
            System.out.println("Enter Your Message: ");
            line = scanner.nextLine();
            dataOutputStream.writeUTF(line);
        }
        System.out.println("Closing Connection ");
```

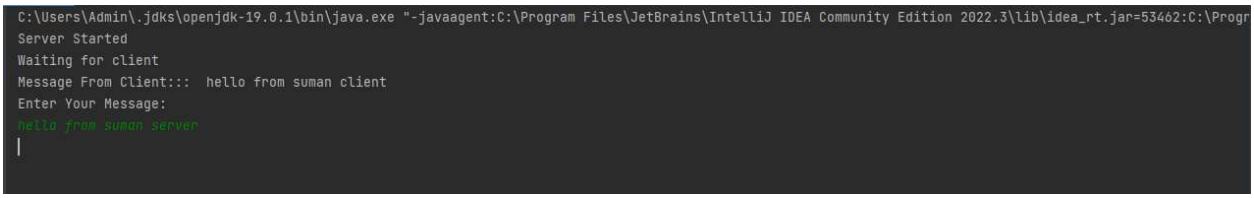
```
        dataInputStream.close();
        serverSocket.close();
    }
}
```

OUTPUT:

Client



Server



RMI

LAB 12

TITLE: Write a Java programs using RMI to find product of two numbers.

AdderClient:

```
package RMI;
import java.rmi.Naming;
public class AdderClient {
	public static void main(String[] args)throws Exception {
	   //	returns the reference of the remote object.
	   AdderInterface adderInterface=(AdderInterface)
	         Naming.lookup("rmi://localhost:8000/add");
	   System.out.println(adderInterface.add(3,3));
	}
   }
```

AdderInterface:

```
package RMI;
import java.rmi.Remote;
import java.rmi.RemoteException;
//For creating the remote interface, extend the Remote interface
public interface AdderInterface extends Remote {
    public int add(int x,int y) throws RemoteException;
}
```

AdderRemote:

```
package RMI;
import java.rmi.RemoteException;
```

```
import java.rmi.server.UnicastRemoteObject;
/* UnicastRemoteObject is used for exporting a remote object with
 Java Remote Method Protocol
  and obtaining a stub that communicates to the remote object.
*/
public class AdderRemote extends UnicastRemoteObject  implements
        AdderInterface {
    AdderRemote() throws RemoteException {
    }
    public int add(int a,int b)
    {
        return a*b;
    }
}
AdderServer
package RMI;
import java.rmi.Naming;
import java.rmi.registry.LocateRegistry;
public class AdderServer {
    public static void main(String[] args) {
        try{
            AdderInterface adderInterface= new AdderRemote();
//            start the registry service giving the port number.
            LocateRegistry.createRegistry(8000);
//            binds the remote object with the given name.
            Naming.bind("rmi://localhost:8000/add",adderInterface);
```

```
            System.out.println("Server Started");
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

Output: