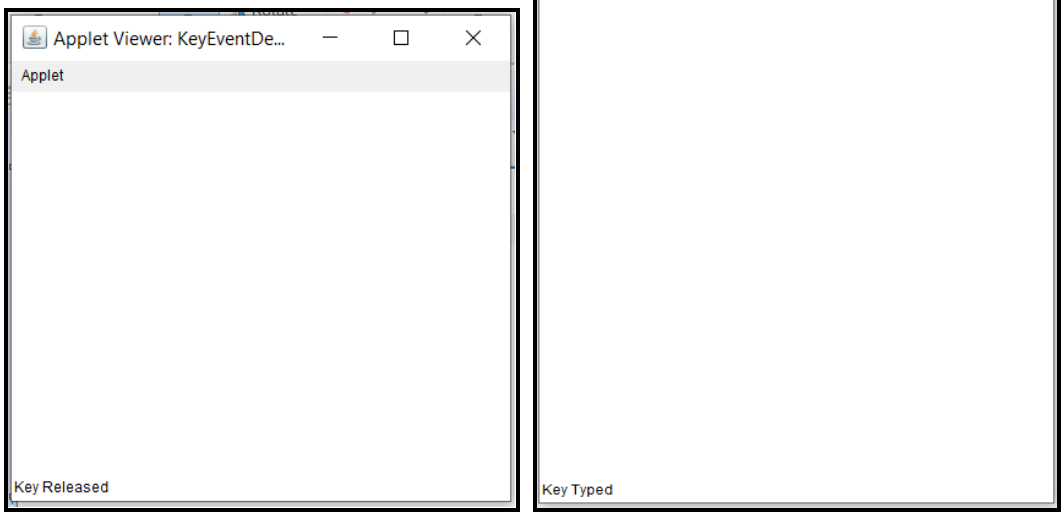
EXPERIMENT NO. 1

Aim: Write a program to demonstrate status of key on Applet window such as KeyPressed, KeyReleased, KeyUp, KeyDown

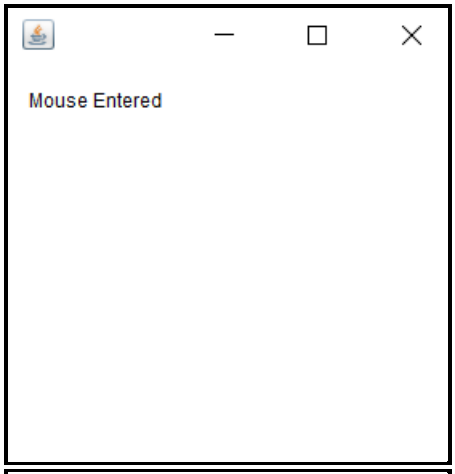
```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class KeyEventDemo extends Applet implements KeyListener
{
   String msg = "";

   public void init()
   {
      addKeyListener(this);
   }

   public void keyReleased(KeyEvent k)
   {
      showStatus("Key Released");
      repaint();
   }

   public void keyTyped(KeyEvent k)
   {
      showStatus("Key Typed");
      repaint();
   }

   public void keyPressed(KeyEvent k)
   {
      showStatus("Key Pressed");
      repaint();
   }

   public void paint(Graphics g)
   {
      g.drawString(msg, 10, 10);
   }
}


/*
  <applet code="KeyEventDemo" height="400" width="400">
  </applet>
*/
```

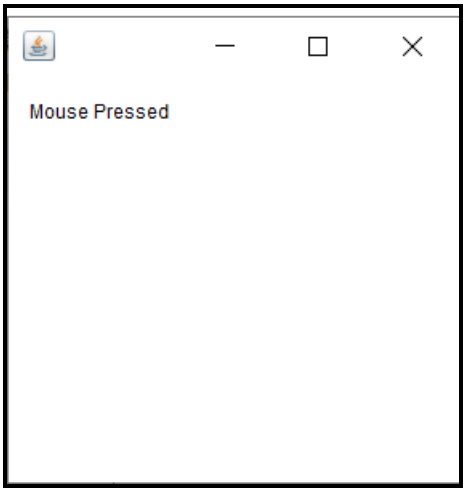

Applet Viewer: KeyEventDe...
Applet
Key Released
Applet Viewer: KeyEventDe...
Applet
Key Typed

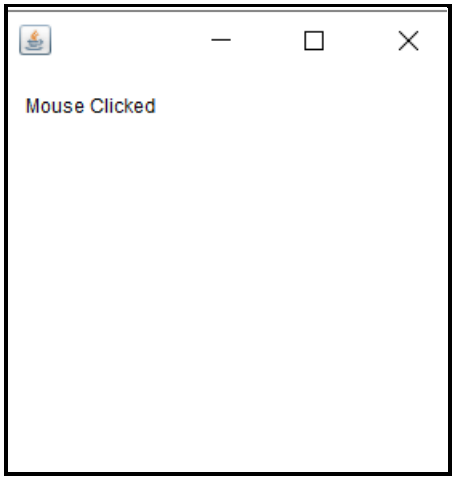
EXPERIMENT NO. 2

Aim: Write a program to create a frame using AWT. Implement mouseClicked, mouseEntered() and mouseExited() events. Frame should become visible when the mouse enters it.

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


public class MouseListenerExample extends Frame implements MouseListener{
   Label l;

     MouseListenerExample(){
       addMouseListener(this);
       l=new Label();
       l.setBounds(20,50,100,20);
       add(l);
       setSize(300,300);
       setLayout(null);
       setVisible(true);
   }
   public void mouseClicked(MouseEvent e) {
       l.setText("Mouse Clicked");
   }
   public void mouseEntered(MouseEvent e) {
       l.setText("Mouse Entered");
   }
   public void mouseExited(MouseEvent e) {
       l.setText("Mouse Exited");
       setVisible(false);
   }
   public void mousePressed(MouseEvent e) {
       l.setText("Mouse Pressed");
   }
   public void mouseReleased(MouseEvent e) {
       l.setText("Mouse Released");
   }
public static void main(String[] args) {
   new MouseListenerExample();
}
}
```


Mouse Entered


Mouse Pressed

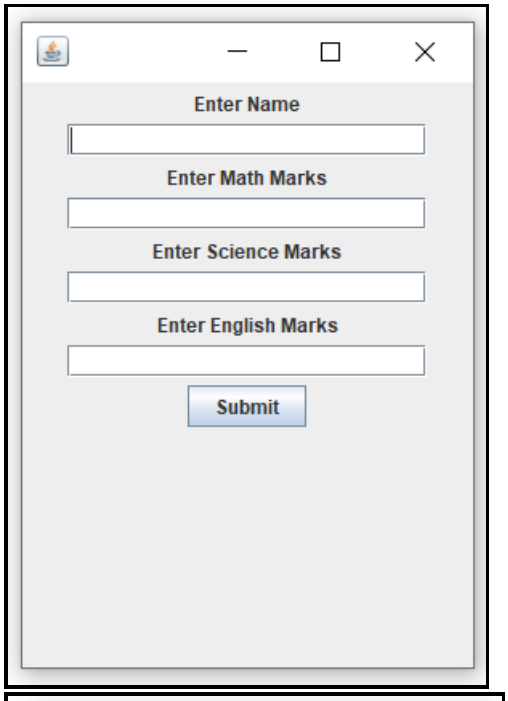

Mouse Clicked

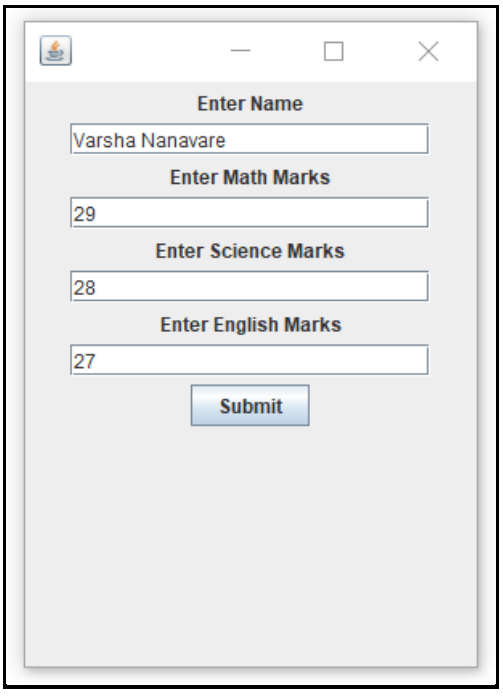
EXPERIMENT NO. 3

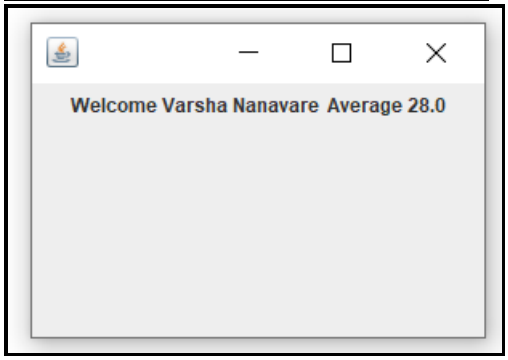
Aim: Develop a GUI which accepts the information regarding the marks for all the subjects of a student in the examination. Display the result for a student in a separate window.

```
import java.awt.BorderLayout;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class ReportCard extends JFrame{
	JPanel jp = new JPanel();
	JLabel lname = new JLabel();
	JButton bsubmit = new JButton("Submit");
	JTextField tname = new JTextField(20);
	JLabel lMath = new JLabel();
	JTextField tMath = new JTextField(20);
	JLabel lScience = new JLabel();
	JTextField tScience = new JTextField(20);
	JLabel lEnglish = new JLabel();
	JTextField tEnglish = new JTextField(20);

	public ReportCard()
	{
		lname.setText("Enter Name");
		jp.add(lname);
		jp.add(tname);
		lMath.setText("Enter Math Marks");
		jp.add(lMath);
		jp.add(tMath);
		lScience.setText("Enter Science Marks");
		jp.add(lScience);
		jp.add(tScience);
		lEnglish.setText("Enter English Marks");
		jp.add(lEnglish);
		jp.add(tEnglish);
		jp.add(bsubmit);
		add(jp);
		bsubmit.addActionListener(new ActionListener(){
			public void actionPerformed(ActionEvent arg0) {
				String val = tname.getText();
				JLabel l1 = new JLabel("Welcome "+val);
				int sub1 = Integer.parseInt(tMath.getText());
				int sub2 = Integer.parseInt(tScience.getText());
				int sub3 = Integer.parseInt(tEnglish.getText());
```

```
                        int sum = sub1+sub2+sub3;
                        float average = sum/3;
                        JLabel l2 = new JLabel("Average "+ average);
                        JPanel jip = new JPanel();
                        jip.add(l1);
                        jip.add(l2);
                        JFrame inf = new JFrame();
                        inf.setVisible(true);
                        inf.add(jip);
                        inf.setSize(300, 100);
                    }

                });

        }

        public static void main(String[] args) {
                ReportCard rc = new ReportCard();
                rc.setSize(300, 200);
                rc.setVisible(true);

        }

}
```





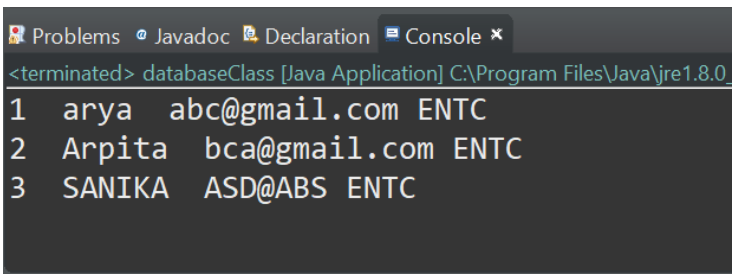


EXPERIMENT NO. 4 – A

Aim: Write a program to insert the data from the database using JDBC create table for studentinfo (rollno int(5) primary key, student name varchar(10), emailid varchar(10), department name varchar(20)).

```
//Java Database Connectivity with MySQL

import java.sql.*;
class databaseClass{
public static void main(String args[]){
try{
//step1 load the driver class
Class.forName("com.mysql.cj.jdbc.Driver");

//step2 create  the connection object  (establish the connection between java
application with oracle database
Connection con=DriverManager.getConnection(
"jdbc:mysql://localhost:3306/student","root","1234");
//here student is database name, root is username and password
      //step3 create the statement object
Statement stmt=con.createStatement();
      //step4 write and execute query
//String query ="create table studentinfo(rollno int(5) primary key, emailid
varchar(10), dept varchar(20))";
//stmt.executeUpdate(query);
//System.out.println("Student data created sussessfuly");


ResultSet rs=stmt.executeQuery("select * from studentinfo");
while(rs.next())
System.out.println(rs.getInt(1)+"  "+rs.getString(2)+"  "+rs.getString(3)+"
"+rs.getString(4));

//step5 close the connection object
con.close();
}catch(Exception e){ System.out.println(e);}
}
}
```

Output :



EXPERIMENT NO. 4 - B

Aim: Write a program to insert and retrieve the data from the database using JDBC.

a) connection. Java file

```
// Java Program to Insert Details in a Table using JDBC
// Connections class

// Importing all SQL classes
// Showing linking of created database

import java.sql.*;

public class connection {

    // object of Connection class
    // initially assigned NULL
    Connection con = null;

    public static Connection connectDB()

    {

        try {

            // Step 2 is involved among 7 in Connection
            // class i.e Load and register drivers

            // 2(a) Loading drivers using forName() method
            // name of database here is mysql
            Class.forName("com.mysql.cj.jdbc.Driver");

            // 2(b) Registering drivers using DriverManager
            Connection con = DriverManager.getConnection(
                "jdbc:mysql://localhost:3306/college",
                "root", "1234");
            // For college Database here (custom sets)
            // root is the username, and
            // 1234 is the password

            // returning the object of Connection class
            // to be used in main class (Example2)
            return con;
        }

        // Catch block to handle the exceptions
        catch (SQLException | ClassNotFoundException e) {

            // Print the exceptions
            System.out.println(e);
```

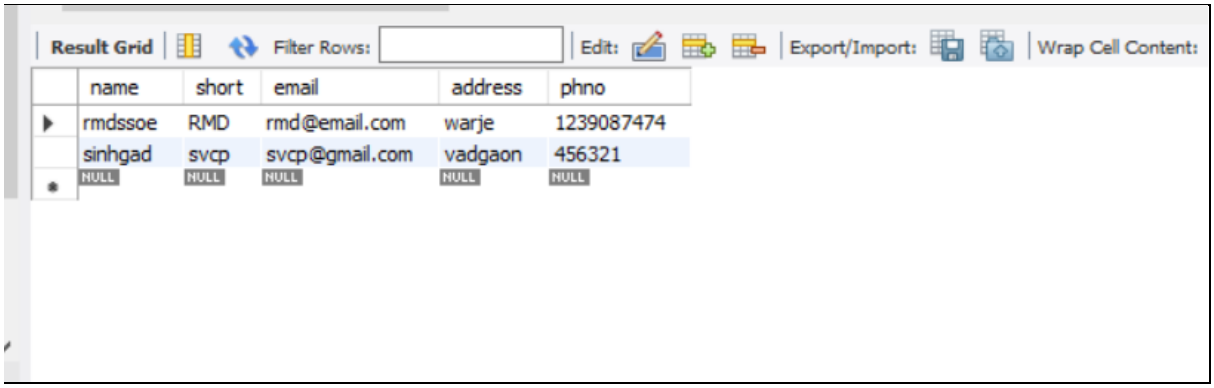
```
            return null;
        }
    }
}
```

b)GFG.java file

```
// Java Program retrieving contents of

// Table Using JDBC connection

 // Java code producing output which is based

// on values stored inside the "collegedata" table in DB


// Java Program to Insert Details in a Table using JDBC and retrieve/ display data
// Main class

// Step 1: Importing DB classes
// DB is SQL here
import java.sql.*;

// Main/App class of above Connection class
public class GFG {

    // MAin driver method
    public static void main(String[] args)
    {
        // Step 2: Showing above Connection class i.e
        // loading and registering drivers

        // Initially assigning NULL parameters
        // to object of Connection class
        Connection con = null;
        PreparedStatement ps = null;

        // Step 3: Establish the connection
        con = connection.connectDB();

        // Try block to check if exception/s occurs
        try {

            // Step 4: Create a statement
            String sql = "insert into  collegedata
values('rmdssoe','RMD','rmd@email.com','warje','1239087474')";

            // Step 5: Execute the query
            ps = con.prepareStatement(sql);

            // Step 6: Process the results
            ps.execute();
        }

        // Optional but recommended
```

```
        // Step 7: Close the connection

        // Catch block to handle the exception/s
        catch (Exception e) {

            // Print the exception
            System.out.println(e);
        }
    }
}
```

Output:



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

name	short	email	address	phno
rmdssoe	RMD	rmd@email.com	warje	1239087474
sinhgad	svcp	svcp@gmail.com	vadgaon	456321
NULL	NULL	NULL	NULL	NULL

EXPERIMENT NO. 5 – A

Aim: Develop an RMI application which accepts a string and checks that string is palindrome or not.

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

    String str = "Radar", reverseStr = "";

    int strLength = str.length();

    for (int i = (strLength - 1); i >=0; --i) {
      reverseStr = reverseStr + str.charAt(i);
    }

    if (str.toLowerCase().equals(reverseStr.toLowerCase())) {
      System.out.println(str + " is a Palindrome String.");
    }
    else {
      System.out.println(str + " is not a Palindrome String.");
    }
  }
}
```

Output

```
Radar is a Palindrome String.
```

EXPERIMENT NO. 5 - B

Aim: Develop an RMI application which accepts a number and checks that number is palindrome or not.

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

    int num = 3553, reversedNum = 0, remainder;

    // store the number to originalNum
    int originalNum = num;

    // get the reverse of originalNum
    // store it in variable
    while (num != 0) {
      remainder = num % 10;
      reversedNum = reversedNum * 10 + remainder;
      num /= 10;
    }

    // check if reversedNum and originalNum are equal
    if (originalNum == reversedNum) {
      System.out.println(originalNum + " is Palindrome.");
    }
    else {
      System.out.println(originalNum + " is not Palindrome.");
    }
  }
}
```

Output

3553 is Palindrome

EXPERIMENT NO. 6

Aim: A Program to demonstrate the use of InetAddress class and its factory methods.

In this program we use InetAddress class and get ip address of www.Facebook.com website.

InetDemo.java

```
import java.io.*;
import java.net.*;
public class InetDemo{
public static void main(String[] args){
try{
InetAddress ip=InetAddress.getByName("www.Facebook.com");

System.out.println("Host Name: "+ip.getHostName());
System.out.println("IP Address: "+ip.getHostAddress());
}catch(Exception e){System.out.println(e);}
}
}
```

Output:

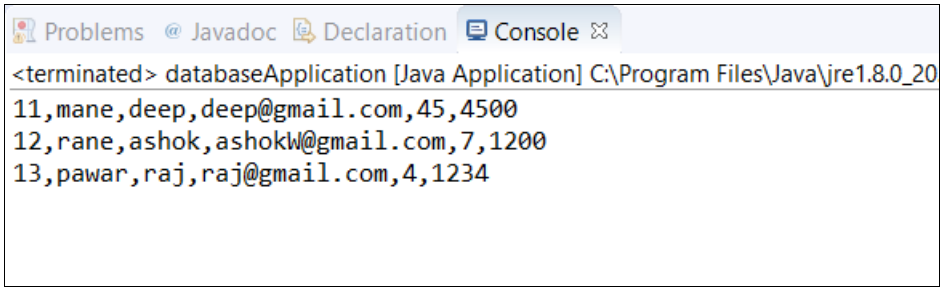
```
Host Name: www.facebook.com
IP Address: 31.13.79.35
```

EXPERIMENT NO. 7

Aim: Write a database application that uses any JDBC driver.

```
// A database application that uses any JDBC driver.

// import sql package to use it in our program
import java.sql.*;

public class databaseApplication {

public static void main(String[] args) throws ClassNotFoundException, SQLException
{
    // store the SQL statement in a string
    String QUERY = "select * from employee_details";
    //register the oracle driver with DriverManager
    Class.forName("com.mysql.cj.jdbc.Driver");
    //Here we have used Java 8 so opening the connection in try statement
    try(Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/employee","root","1234"))
    {
            Statement statemnt1 = conn.createStatement();
            //Created statement and execute it
            ResultSet rs1 = statemnt1.executeQuery(QUERY);
            {
                //Get the values of the record using while loop
                while(rs1.next())
                {
                    int empNumber = rs1.getInt("empNumber");
                    String lastName = rs1.getString("lastName");
                    String firstName = rs1.getString("firstName");
                    String email = rs1.getString("email");
                    String deptNum = rs1.getString("deptNum");
                    String salary = rs1.getString("salary");
                    //store the values which are retrieved using ResultSet and
print it
                System.out.println(empNumber + "," +lastName+ "," +firstName+ ","
+email +","+deptNum +"," +salary);
                }
            }
    }
    catch (SQLException e) {
        //If exception occurs catch it and exit the program
        e.printStackTrace();
    }
   }
}
```

Output:


Problems
Javadoc
Declaration
Console
<terminated> databaseApplication [Java Application] C:\Program Files\Java\jre1.8.0_20
11,mane,deep,deep@gmail.com,45,4500
12,rane,ashok,ashokW@gmail.com,7,1200
13,pawar,raj,raj@gmail.com,4,1234

EXPERIMENT NO. 8

Aim: Write program with suitable example to develop your remote interface, implement you're RMI server, implement application that create your server, also develop security policy file.

Defining the Remote Interface

Hello.java

```
import java.rmi.Remote;
import java.rmi.RemoteException;

// Creating Remote interface for our application
public interface Hello extends Remote {
   void printMsg() throws RemoteException;
}
```

Developing the Implementation Class (Remote Object)

ImplExample.java

```
// Implementing the remote interface
public class ImplExample implements Hello {

   // Implementing the interface method
   public void printMsg() {
      System.out.println("This is an example RMI program");
   }
}
```

Developing the Server Program

Server.java

```
import java.rmi.registry.Registry;
import java.rmi.registry.LocateRegistry;
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;

public class Server extends ImplExample {
   public Server() {}
   public static void main(String args[]) {
```

```
      try {
         // Instantiating the implementation class
         ImplExample obj = new ImplExample();

         // Exporting the object of implementation class
         // (here we are exporting the remote object to the stub)
         Hello stub = (Hello)
UnicastRemoteObject.exportObject(obj, 0);

         // Binding the remote object (stub) in the registry
         Registry registry = LocateRegistry.getRegistry();

         registry.bind("Hello", stub);
         System.err.println("Server ready");
      } catch (Exception e) {
         System.err.println("Server exception: " + e.toString());
         e.printStackTrace();
      }
   }
}
```

Developing the Client Program

Client.java

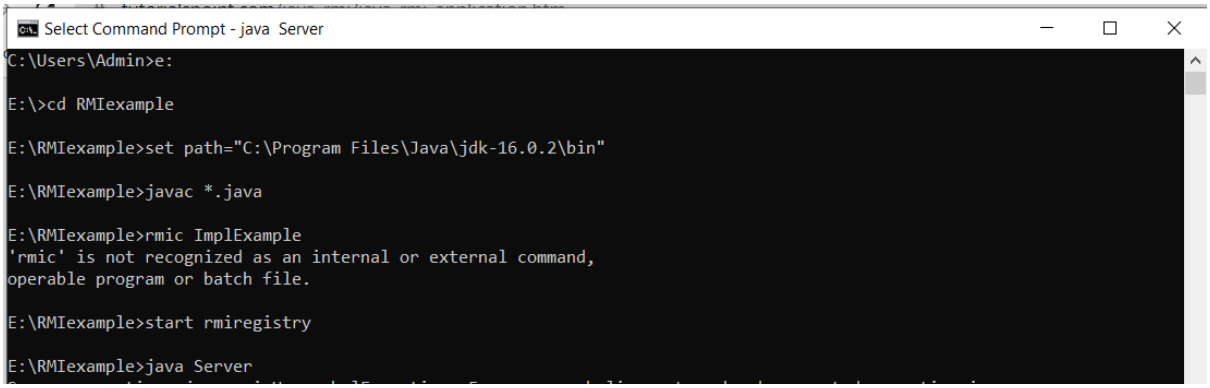
```
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

public class Client {
   private Client() {}
   public static void main(String[] args) {
      try {
         // Getting the registry
         Registry registry = LocateRegistry.getRegistry(null);

         // Looking up the registry for the remote object
         Hello stub = (Hello) registry.lookup("Hello");

         // Calling the remote method using the obtained object
         stub.printMsg();

         // System.out.println("Remote method invoked");
      } catch (Exception e) {
         System.err.println("Client exception: " + e.toString());
         e.printStackTrace();
      }
   }
}
```

Compiling the Application

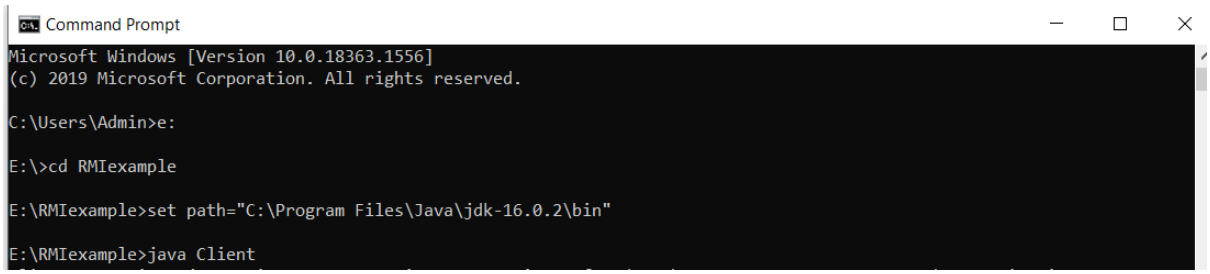
```
Javac *.java
```

Screenshot's of Output:

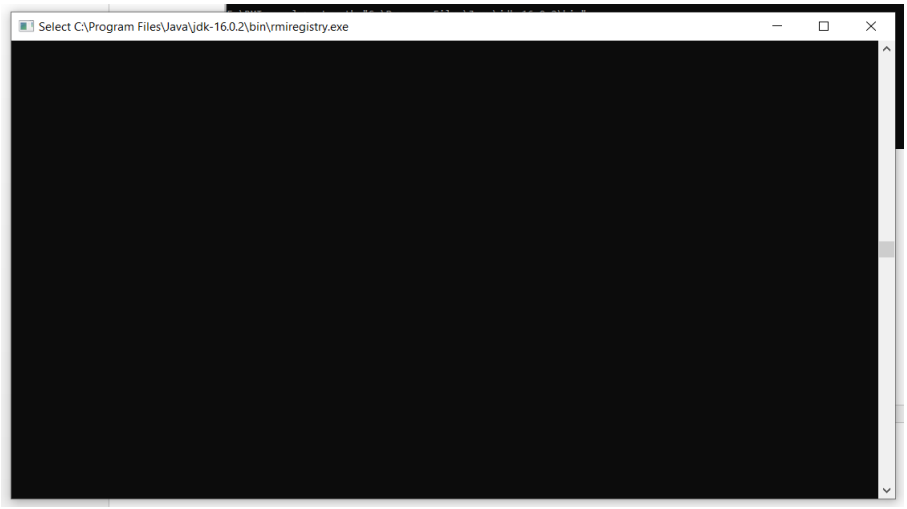
a) Server



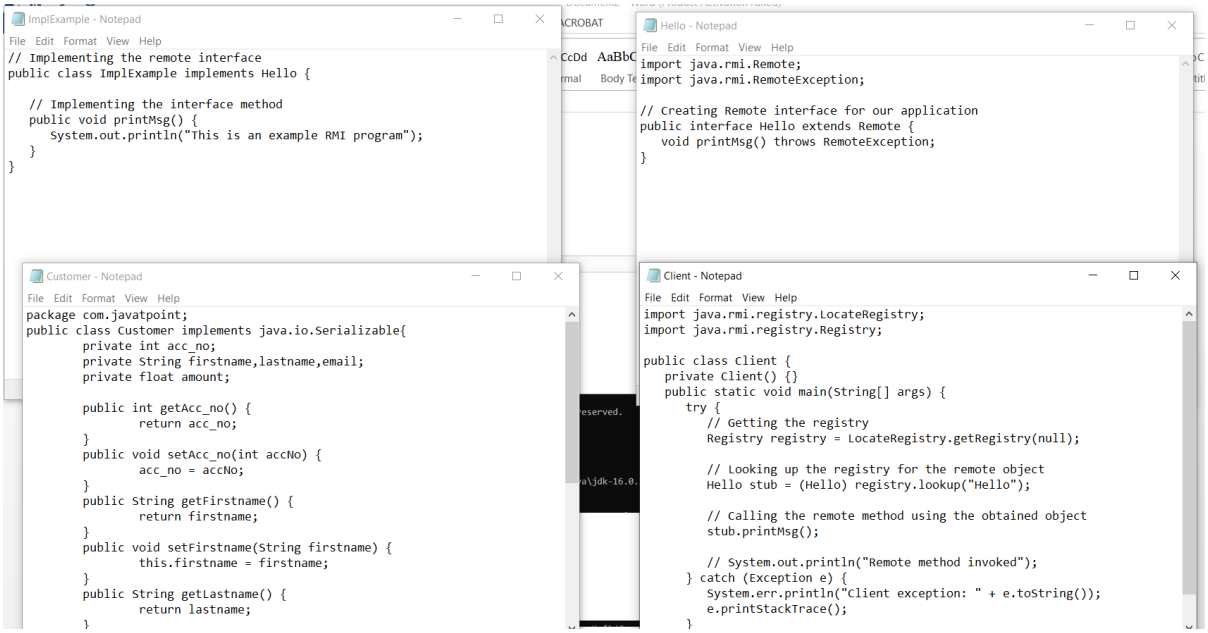
b) Client



c) Rmiregistry



Program screenshots:



ImplExample - Notepad

```
// Implementing the remote interface
public class ImplExample implements Hello {

   // Implementing the interface method
   public void printMsg() {
      System.out.println("This is an example RMI program");
   }
}
```

Hello - Notepad

```
import java.rmi.Remote;
import java.rmi.RemoteException;

// Creating Remote interface for our application
public interface Hello extends Remote {
   void printMsg() throws RemoteException;
}
```

Customer - Notepad

```
package com.javatpoint;
public class Customer implements java.io.Serializable{
        private int acc_no;
        private String firstname,lastname,email;
        private float amount;

        public int getAcc_no() {
                return acc_no;
        }
        public void setAcc_no(int accNo) {
                acc_no = accNo;
        }
        public String getFirstname() {
                return firstname;
        }
        public void setFirstname(String firstname) {
                this.firstname = firstname;
        }
        public String getLastname() {
                return lastname;
        }
```

Client - Notepad

```
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

public class Client {
   private Client() {}
   public static void main(String[] args) {
      try {
         // Getting the registry
         Registry registry = LocateRegistry.getRegistry(null);

         // Looking up the registry for the remote object
         Hello stub = (Hello) registry.lookup("Hello");

         // Calling the remote method using the obtained object
         stub.printMsg();

         // System.out.println("Remote method invoked");
      } catch (Exception e) {
         System.err.println("Client exception: " + e.toString());
         e.printStackTrace();
      }
```