TRINITY INTERNATIONAL COLLEGE

(Tribhuvan University Affiliated)

Lab Assignment 8: Advance Java Programming

Submitted By:

Submitted to:

Name:

Program: **B. Sc. (CSIT)**

Roll No:

Semester: seventh (7th)

Date:

KATHMANDU, NEPAL
2020

1. Use RMI to develop programs that runs in different machines. [2070]

⇨

Program

compute.java (common for both client and server)

```
package compute;

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

public interface Compute extends Remote
{
    public double add(double a, double b)throws RemoteException;

}
```

RmiClient.java

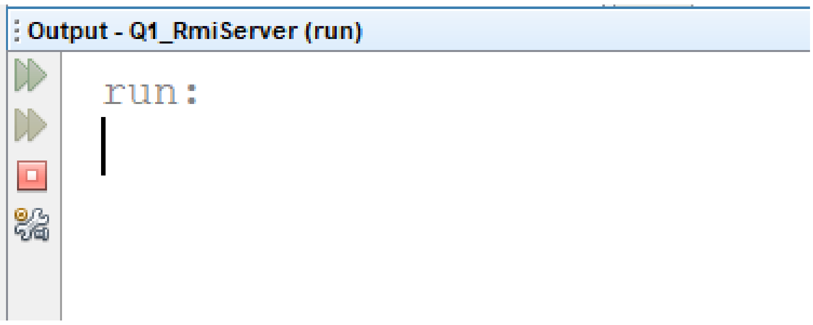
```
package Q1_rmiclient;

import compute.*;
import java.rmi.Naming;

public class RmiClient
{


    public static void main(String[] args)
    {
        try{
            String url = "rmi://127.0.0.1:8888/Compute";
            Compute compute = (Compute)Naming.lookup(url);
            double result = compute.add(5,6);
            System.out.println("Sum = "+ result);
        }
        catch(Exception e)
        {
            System.err.println("Remote exception:");
        }
    }
}
```

RmiServer.java

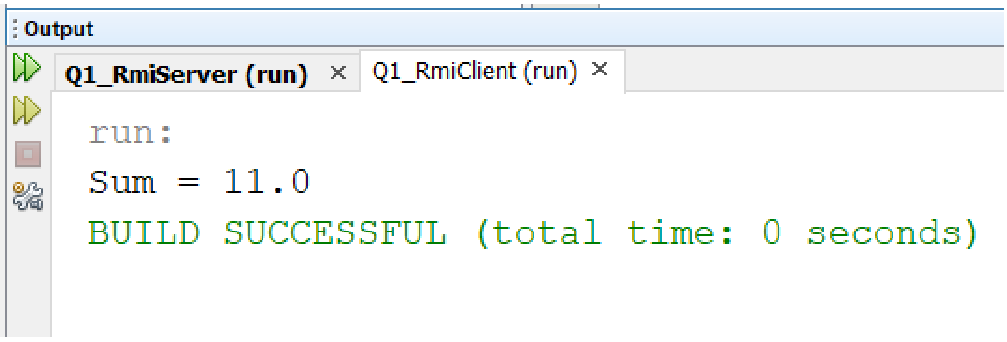
```
package Q1_rmiserver;

import compute.*;
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.rmi.server.UnicastRemoteObject;

public class RmiServer implements Compute
{

     public RmiServer() {super();}
```

```
  public double add (double a , double b)throws RemoteException
    {
        return a+b;
    }

    public static void main(String[] args)
    {
        try
        {
            Compute server = new RmiServer();
            final int PORT = 8888;
            Registry registry = LocateRegistry.createRegistry
                                                            (PORT);
            UnicastRemoteObject.exportObject(server,PORT);
            registry.rebind("Compute", server);

        }catch(Exception e)
        {
            System.err.println("RmiServer Exception.");
        }
    }
}
```

Output:



```
run:
```



```
run:
Sum = 11.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Write distributed programs with client and server using RMI to find the area of a
   a) Circle
   b) Rectangle, and
   c) Sphere

   a) Circle

⇨

Program

Compute.java (Common for both client and server)

```
package compute;

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

public interface Compute extends Remote
{
    public double calcArea(double r)throws RemoteException;

}
```

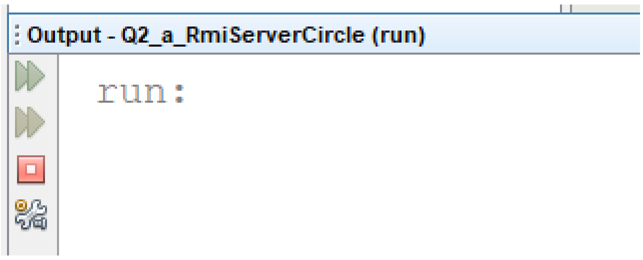
RmiClientCircle.java

```
package q2_a_rmiclientcircle;
import compute.*;
import java.rmi.Naming;


public class RmiClientCircle
{
       public static void main(String[] args)
       {
          try
          {
           String url = "rmi://127.0.0.1:8888/Compute";
           Compute compute = (Compute)Naming.lookup(url);
           double result = compute.calcArea(1);
           System.out.println("Area of Circle = "+ result);
           }
           catch(Exception e)
           {
              System.err.println("Remote exception:");
            }
       }
}
```

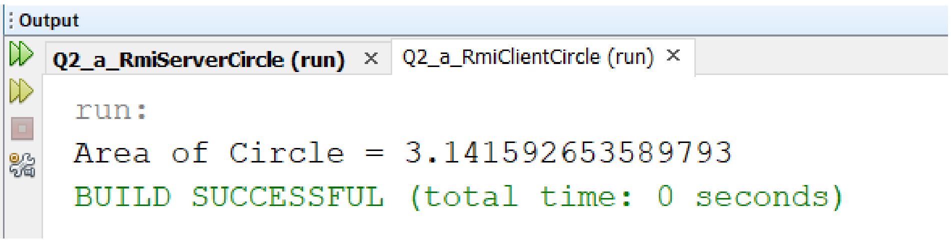
RmiServerCircle.java

```
package q2_a_rmiservercircle;

import compute.*;
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.rmi.server.UnicastRemoteObject;

public class RmiServerCircle implements Compute
{

    public RmiServerCircle() {super();}
    public double calcArea (double r) throws RemoteException
    {
        return Math.PI*r*r;
    }

    public static void main(String[] args)
    {
        try
        {
            Compute server = (Compute) new RmiServerCircle();
            final int PORT = 8888;
            Registry registry = LocateRegistry.createRegistry
                                                          (PORT);
            UnicastRemoteObject.exportObject(server,PORT);
            registry.rebind("Compute", server);

        }catch(Exception e)
        {
            System.err.println("RmiServer Exception.");
        }
    }
}
```

Output:



```
run:
```



```
run:
Area of Circle = 3.141592653589793
BUILD SUCCESSFUL (total time: 0 seconds)
```

b) Rectangle

⇨

Program

Compute.java (common for both client and server)

```
package compute;

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

public interface Compute extends Remote
{
    public double calcArea(double l ,double b) throws
                                                RemoteException;
}
```

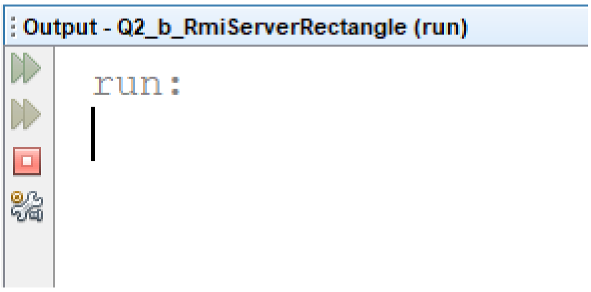
RmiClientRectangle.java

```
package q2_b_rmiclientrectangle;

import compute.Compute;
import java.rmi.Naming;

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

           try{
           String url = "rmi://127.0.0.1:8888/Compute";
           Compute compute = (Compute)Naming.lookup(url);
           double result = compute.calcArea(5,6);
           System.out.println("Area of Rectangle= "+ result);
           }
          catch(Exception e)
          {
           System.err.println("Remote exception:");
          }
    }
}
```

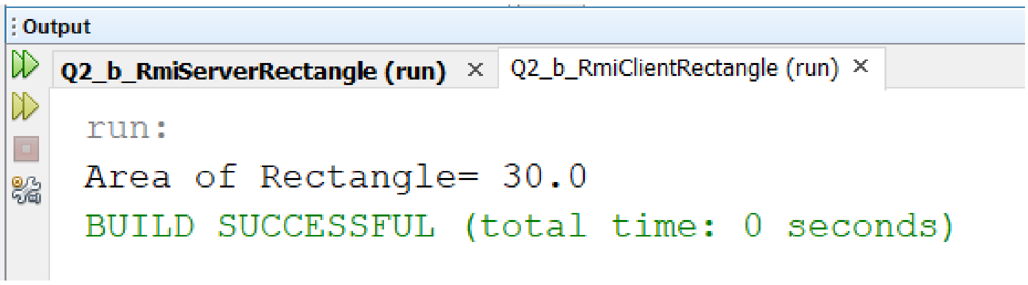
RmiServerRectangle.java

```
package q2_b_rmiserverrectangle;

import compute.Compute;
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.rmi.server.UnicastRemoteObject;

public class RmiServerRectangle implements Compute
{

    public RmiServerRectangle() {super();}
    public double calcArea (double l ,double b) throws
                                              RemoteException
    {
        return l*b;
    }

    public static void main(String[] args)
    {
        try
        {
            Compute server = new RmiServerRectangle();
            final int PORT = 8888;
            Registry registry = LocateRegistry.createRegistry
                                                   (PORT);
            UnicastRemoteObject.exportObject(server,PORT);
            registry.rebind("Compute", server);

        }
        catch(Exception e)
        {
            System.err.println("RmiServer Exception.");

        }
    }

}
```

Output:



```
run:
```



```
run:
Area of Rectangle= 30.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

c) Sphere

⇨

Program

Compute.java (common for both client and server)

```
package compute;

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

public interface Compute extends Remote
{
    public double calcArea(double r) throws RemoteException;

}
```

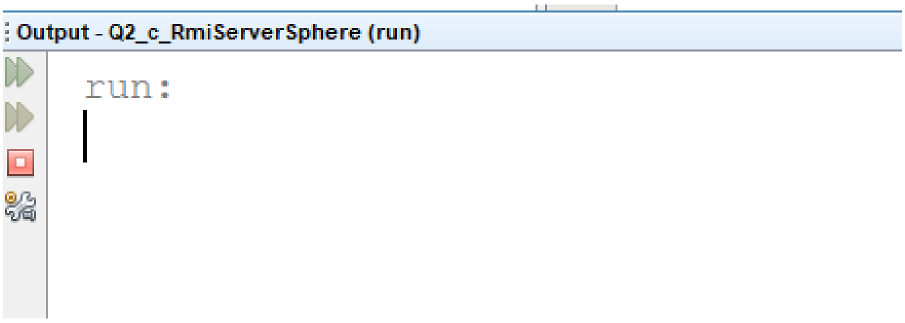
RmiClientSphere.java

```
package q2_c_rmiclientsphere;

import compute.Compute;
import java.rmi.Naming;

public class RmiClientSphere
{
     public static void main(String[] args)
    {
         try{
            String url = "rmi://127.0.0.1:8888/Compute";
            Compute compute = (Compute)Naming.lookup(url);
            double result = compute.calcArea(1);
            System.out.println("Area of Sphere= "+ result);
        }
        catch(Exception e)
        {
            System.err.println("Remote exception:");
        }

    }

}
```

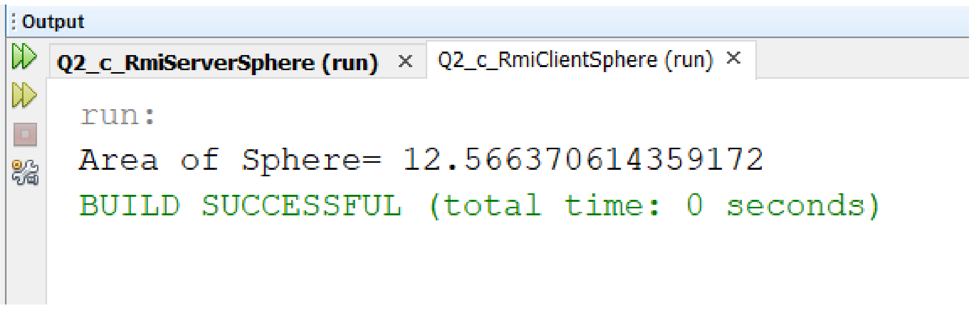
RmiServerSphere.java

```
package q2_c_rmiserversphere;

import compute.Compute;
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.rmi.server.UnicastRemoteObject;

public class RmiServerSphere implements Compute
{
    public RmiServerSphere() {super();}
    public double calcArea (double r) throws RemoteException
    {
        return 4*Math.PI*Math.pow(r, 2);
    }

    public static void main(String[] args)
    {
        try
        {
            Compute server = new RmiServerSphere();
            final int PORT = 8888;
            Registry registry = LocateRegistry.createRegistry
                                                        (PORT);
            UnicastRemoteObject.exportObject(server,PORT);
            registry.rebind("Compute", server);

        }
        catch(Exception e)
        {
            System.err.println("RmiServer Exception.");

        }
    }
}
```

Output:



```
run:
```



```
run:
Area of Sphere= 12.566370614359172
BUILD SUCCESSFUL (total time: 0 seconds)
```