

## Practical No. 03

**Q1. Design a Graphical User Interface (GUI) based calculator. (scientific or standard). Operations should be performed using both mouse and keyboard.**

### Server.java

```
package ishwari;
import java.rmi.Naming; import
java.rmi.registry.LocateRegistry;
public class Server
{
    public static void main(String[] args) {
        try
        {
            System.out.println("Calculator Service Started");
            adder stub= new CalcOperation();
            System.out.println("Calculator Service Binding...");
            LocateRegistry.createRegistry(5000);

            Naming.rebind("rmi://localhost:5000/CalcOpService", stub);
            System.out.println("Calculator Service is registered in
registry");
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

### Client.java

```
package ishwari;
import
java.rmi.Naming; public
class Client {

    public static void main(String[] args) {
        try
        {
            System.out.println("Client Program Started");

            adder stub =
(adder)Naming.lookup("rmi://localhost:5000/CalcOpService");
            System.out.println("Addition is:
```

```

"+stub.getAddition(34,4));
        System.out.println("Subtraction is:
"+stub.getSubtraction(34,4));
    }
    catch(Exception e)
    {
        System.out.println(e);
    }
}
}

```

### CalcOperation.java

```

package ishwari;
import
java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;
public class CalcOperation extends UnicastRemoteObject implements adder
{ private static final long serialVersionUID = 1L;
    CalcOperation() throws RemoteException
    {
        super();
    }
    @Override
    public int getAddition(int num1,int num2) throws RemoteException
    {
        return num1+num2;
    }

    @Override
    public int getSubtraction(int num1,int num2) throws RemoteException
    {
        return num1-num2;
    }
}

```

### adder.java

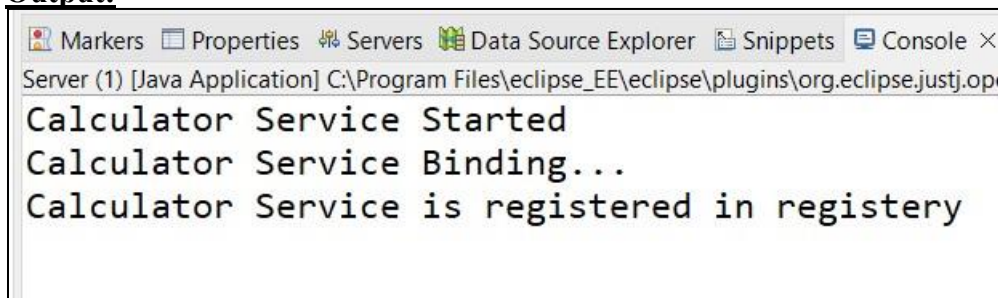
```

package ishwari;

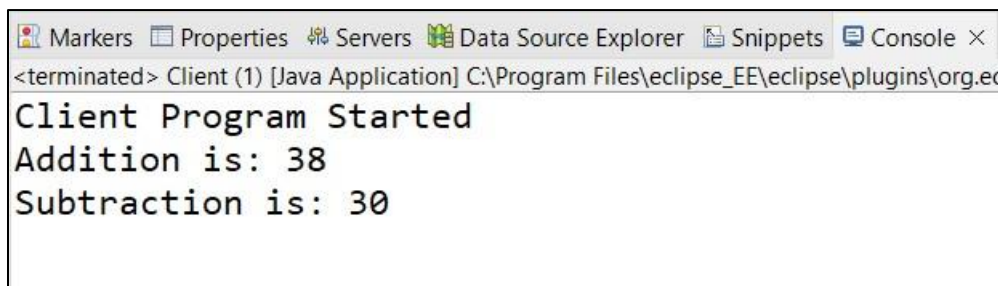
import java.rmi.Remote; import
java.rmi.RemoteException;
public interface adder extends
Remote { public int
getAddition(int num1,int num2)
throws RemoteException;
public int getSubtraction(int num1,int num2)
throws RemoteException;
}

```

### Output:

A screenshot of the Eclipse IDE's console window. The title bar shows 'Markers', 'Properties', 'Servers', 'Data Source Explorer', 'Snippets', and 'Console'. The console text shows the output of a Java application: 'Calculator Service Started', 'Calculator Service Binding...', and 'Calculator Service is registered in registry'.

```
Server (1) [Java Application] C:\Program Files\eclipse_EE\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64.jre\bin\java.exe
Calculator Service Started
Calculator Service Binding...
Calculator Service is registered in registry
```

A screenshot of the Eclipse IDE's console window. The title bar shows 'Markers', 'Properties', 'Servers', 'Data Source Explorer', 'Snippets', and 'Console'. The console text shows the output of a client Java application: '<terminated> Client (1) [Java Application] C:\Program Files\eclipse\_EE\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64.jre\bin\java.exe', 'Client Program Started', 'Addition is: 38', and 'Subtraction is: 30'.

```
<terminated> Client (1) [Java Application] C:\Program Files\eclipse_EE\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64.jre\bin\java.exe
Client Program Started
Addition is: 38
Subtraction is: 30
```

**Q.2. Retrieve day, time and date function from server to client. This program should display server day, date and time.**

**Ans :**

- **Code : DateTimeClient.java**

```
package dateTime;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
public class DateTimeClient { public static void
main(String[] args) throws Exception{
    try {
        Registry registry = LocateRegistry.getRegistry("localhost",
        5050);
        DateTimeService dateTimeService =
        (DateTimeService) registry.lookup("DateTimeService");
        // Invoke remote methods
        String day = dateTimeService.getCurrentDay();
        String time = dateTimeService.getCurrentTime();
        String date = dateTimeService.getCurrentDate();
```

```

System.out.println("Server Day: " + day);
System.out.println("Server Time: " + time);
System.out.println("Server Date: " + date);
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

### **DateTimeServer.java**

```

package dateTime;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.rmi.server.UnicastRemote
Object ;
public class DateTimeServer {
public static void main(String[] args)
{ try {
DateTimeService dateTimeService = new
DateTimeServiceImpl();
DateTimeService stub = (DateTimeService)
UnicastRemoteObject.exportObject(dateTimeSe
rvice, 0); Registry registry =
LocateRegistry.createRegistry(5050);
registry.rebind("DateTimeService", stub);
System.out.println("Server is running...");
} catch
(Exception e) { e.printStackTrace(); }
}
}

```

### **DateTimeService.java**

```
package dateTime;
import java.rmi.Remote;
import java.rmi.RemoteException;
interface DateTimeService extends Remote
{
String getCurrentDay() throws
RemoteException;
String getCurrentTime() throws RemoteException;
String getCurrentDate() throws RemoteException;
}
```

### **DateTimeServiceImpl.java**

```
package dateTime;
import java.rmi.RemoteException;
import java.text.SimpleDateFormat;
import java.util.Date;
class DateTimeServiceImpl implements
DateTimeService {
@Override public String getCurrentDay() throws
RemoteException {
SimpleDateFormat sdf = new
SimpleDateFormat("EEEE");
return sdf.format(new Date());
}
@Override
public String getCurrentTime() throws RemoteException {
SimpleDateFormat sdf = new
SimpleDateFormat("HH:mm:ss"); return sdf.format(new
Date());
}
```

@Override

```
public String getCurrentDate() throws RemoteException {  
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MMdd");  
    return sdf.format(new Date());  
}  
}
```

• **Output :**

