Practical No.2

Objective: Calculator using RMI. **Theory:**

Steps to Implement RMI: -

1. Create a Remote interface - extending java.rmi Remote interface
2. Implementing Remote interface- by making a server class
3. Write server program - Create the objects of the server class and register them in the rmi registry
4. Write the client program - access the remote server object, through the golbal name in the regsitry
5. Compiling and creating stub and skeleton - by rmic tool
6. Start the rmi registry service
7. Run the server program
8. Run the client program - Running on separate command window
9. javac *.java
10. rmic CalcImp
11. start rmiregistry
12. java Server
12. java Client

Source Code:

Calcimp.java import

```
java.rmi.server.UnicastRemoteObject;
public class Calcimp extends UnicastRemoteObject implements Calc
{
   public Calcimp()throws Exception
   {
      super();

   }
   public int calc(int x,int y,char ch)
{
      switch(ch)
      {
         case '+': return(x+y);
case '-': return(x-y);
case 'x': return(x*y);
case '/': return(x/y);
default : return 0;
      }
   }
}
```

Server.java import

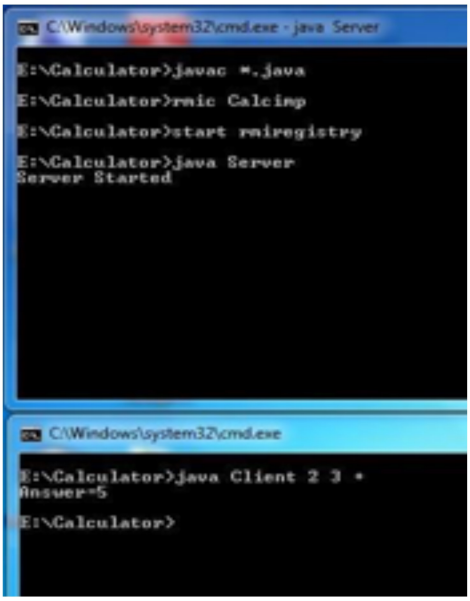
```
java.rmi.*;
public class Server extends Calcimp
{
   public Server() throws Exception
```

```
    {}
            public static void main (String args[]) throws Exception
    {
try
        {
            Calcimp ob = new Calcimp();
            Naming.bind("A",ob);
            System.out.println("Server Started");
                }
        catch(Exception e)
        {
            System.out.println("Server Exception"+e);
        }
    }
}
```

Client.java import

```
java.rmi.*; public
class Client
{
    public Client(){}
    public static void main(String args[]) throws Exception
    {
        int num1,num2;
char ch;
        Calc ob = (Calc)Naming.lookup("A");
        num1 = Integer.parseInt(args[0]);
num2 = Integer.parseInt(args[1]);
ch= args[2].charAt(0);                int c=
ob.calc(num1,num2,ch);
System.out.println("Answer="+c);
    }
}
```

Output:



Result: The given program has been compiled and executed successfully