Laboratory Manual

For

Distributed Computing

M. Tech. (IT)

SEM I

June 2011

Faculty of Technology
Dharmsinh Desai University
Nadiad.
www.ddu.ac.in

LAB-1

AIM: Implement a program of RPC (Remote Procedure Call)

(to calculate area of triangle)

area-tri.x

```
struct intpair {
        float height;
        float base;
        };

    program MATHPROG {
        version MATHVERS {
            float area(intpair) = 1;

        } = 1;
    } = 0x20000008;
```

SERVER CODE – area-tri_server.c

```
/*
 * This is sample code generated by rpcgen.
 * These are only templates and you can use them
 * as a guideline for developing your own functions.
 */

#include "area-tri.h"

float *
area_1_svc(intpair *argp, struct svc_req *rqstp)
{
        static float  ans;

        ans = 0.5*argp->height*argp->base;
        return &ans;
}
```

CLIENT CODE – area-tri_client.c

```
/ * This is sample code generated by rpcgen.
 * These are only templates and you can use them
 * as a guideline for developing your own functions.*/
#include "area-tri.h"
void mathprog_1(char *host)
{       CLIENT *clnt;
        float  *result_1;
        intpair  area_1_arg;
#ifndefDEBUG
        clnt = clnt_create (host, MATHPROG, MATHVERS, "udp");
        if (clnt == NULL) {
                clnt_pcreateerror (host);
                exit (1);           }
#endif /* DEBUG */
        result_1 = area_1(&area_1_arg, clnt);
        if (result_1 == (float *) NULL) {
                clnt_perror (clnt, "call failed");              }
#ifndefDEBUG
        clnt_destroy (clnt);
#endif  /* DEBUG */   }

int main (int argc, char *argv[])
{       char *host;
        intpair s;
        CLIENT *cl;
        float *ans;
        if (argc < 2) {
                printf ("usage: %s server_host\n", argv[0]);
                exit (1);           }
        cl= clnt_create(argv[1], MATHPROG , MATHVERS ,"udp");
        printf("\nEnter Height:");
        scanf("%f",&s.height);
        printf("\nEnter Base:");
        scanf("%f",&s.base);
        ans = area_1(&s, cl);
        printf("Area of Triangle is : %f\n", *ans);
exit (0);
}
```

OUTPUT:

```
user1@MTechLab:~/Desktop/DC-La
File  Edit  View  Search  Terminal  Help
[user1@MTechLab RPC]$ rpcgen -a area-tri.x
[user1@MTechLab RPC]$ make -f Makefile.area-tri
cc -g    -c -o area-tri_clnt.o area-tri_clnt.c
In file included from area-tri_clnt.c:7:
area-tri.h:6:14: warning: extra tokens at end of #ifndef directive
area-tri.h:7:14: warning: missing whitespace after the macro name
cc -g    -c -o area-tri_client.o area-tri_client.c
In file included from area-tri_client.c:7:
area-tri.h:6:14: warning: extra tokens at end of #ifndef directive
area-tri.h:7:14: warning: missing whitespace after the macro name
cc -g    -c -o area-tri_xdr.o area-tri_xdr.c
In file included from area-tri_xdr.c:6:
area-tri.h:6:14: warning: extra tokens at end of #ifndef directive
area-tri.h:7:14: warning: missing whitespace after the macro name
cc -g     -o area-tri_client  area-tri_clnt.o area-tri_client.o area-tri_xdr.o -
lnsl
cc -g    -c -o area-tri_svc.o area-tri_svc.c
In file included from area-tri_svc.c:6:
area-tri.h:6:14: warning: extra tokens at end of #ifndef directive
area-tri.h:7:14: warning: missing whitespace after the macro name
cc -g    -c -o area-tri_server.o area-tri_server.c
In file included from area-tri_server.c:7:
area-tri.h:6:14: warning: extra tokens at end of #ifndef directive
area-tri.h:7:14: warning: missing whitespace after the macro name
cc -g     -o area-tri_server  area-tri_svc.o area-tri_server.o area-tri_xdr.o -l
nsl
[user1@MTechLab RPC]$ ./area-tri_server
^C
[user1@MTechLab RPC]$
```

```
user1@MTechLab:~/Desktop/DC-Labs_BDR/LAB-1-RPC/RPC
File  Edit  View  Search  Terminal  Help
[user1@MTechLab RPC]$ ./area-tri_client 192.168.34.23

Enter Height:4

Enter Base:9
Area of Triangle is : 18.000000
[user1@MTechLab RPC]$
```

LAB-2

AIM: Implement the program of RMI (Remote Method Invocation) (to calculate area of triangle)

AreaServerIntf.java - Remote Interface

```
import java.rmi.*;

public interface AreaServerIntf extends Remote
{
  double area_triangle(double d1, double d2) throws RemoteException;
}
```

AreaServerImpl.java - Implements the Remote Interface

```
import java.rmi.*;
import java.rmi.server.*;

public class AreaServerImpl extends UnicastRemoteObject  implements
AreaServerIntf
{
  public AreaServerImpl() throws RemoteException
   {
    }
  public double area_triangle(double d1, double d2) throws RemoteException
  {
    return (0.5*d1*d2);
  }
}
```

AreaServer.java

```
import java.net.*;
import java.rmi.*;

public class AreaServer {
  public static void main(String args[]) {
    try {
      AreaServerImpl areaServerImpl = new AreaServerImpl();
      Naming.rebind("AreaServer", areaServerImpl);
    }
    catch(Exception e) {
      System.out.println("Exception: " + e);
    }
  }
}
```

AreaClient.java

```
import java.rmi.*;
public class AreaClient {
  public static void main(String args[]) {
    try {
      String areaServerURL = "rmi://" + args[0] + "/AreaServer";
      AreaServerIntf areaServerIntf =
                (AreaServerIntf)Naming.lookup(areaServerURL);
      System.out.println("Height is: " + args[1]);
      double d1 = Double.valueOf(args[1]).doubleValue();
      System.out.println("Base is: " + args[2]);
      double d2 = Double.valueOf(args[2]).doubleValue();
      System.out.println("The area of triangle is: " + areaServerIntf.area_triangle(d1,
      d2));
    }
    catch(Exception e) {
              System.out.println("Exception: " + e);
    }
  }
}
```

OUTPUT:

```
C:\Windows\System32\cmd.exe - java AreaServer
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

D:\DC-Labs_BDR\LAB-2-RMI\RMI-triangle>javac *.java

D:\DC-Labs_BDR\LAB-2-RMI\RMI-triangle>start rmiregistry

D:\DC-Labs_BDR\LAB-2-RMI\RMI-triangle>rmic AreaServerImpl

D:\DC-Labs_BDR\LAB-2-RMI\RMI-triangle>java AreaServer
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

D:\DC-Labs_BDR\LAB-2-RMI\RMI-triangle>java AreaClient localhost 8 4
Height is: 8
Base is: 4
The area of triangle is: 16.0

D:\DC-Labs_BDR\LAB-2-RMI\RMI-triangle>_
```

LAB-3

AIM: Implement the program for CORBA (to calculate area of triangle)

Write the IDL specification of the interface methods, for example:

area.idl

```
module areaApp
{
  interface area
  {
    double area_tri(in double H,in double B);
    oneway void shutdown();
    };
};
```

Compile the IDL specification:

- **idlj –fall area.idl**

The –fall command option is necessary for the compiler to generate all the files needed. As a result, a subdirectory HelloApp will be created including a number of Java files, with the following names:

- areaPOA.java
- _areaStub.java
- area.java
- areaHelper.java
- areaHolder.java
- areaOperations.java

areaObj.java

```
import areaApp.*;
import org.omg.CosNaming.*;
import org.omg.CosNaming.NamingContextPackage.*;
import org.omg.CORBA.*;
import org.omg.PortableServer.*;
import org.omg.PortableServer.POA;
import java.util.Properties;
class areaObj extends areaPOA {
  private ORB orb;
  public void setORB(ORB orb_val) {
    orb = orb_val;
  }
  // implement area_tri() method
  public double area_tri(double Height,double Base) {
    double ans=(0.5*Height*Base);
    return ans;
  }
  // implement shutdown() method
  public void shutdown() {
    orb.shutdown(false);
  }
}
```

StartClient.java

```
import areaApp.*;
import org.omg.CosNaming.*;
import org.omg.CosNaming.NamingContextPackage.*;
import org.omg.CORBA.*;
import java.io.*;
import java.util.*;

public class StartClient {
```

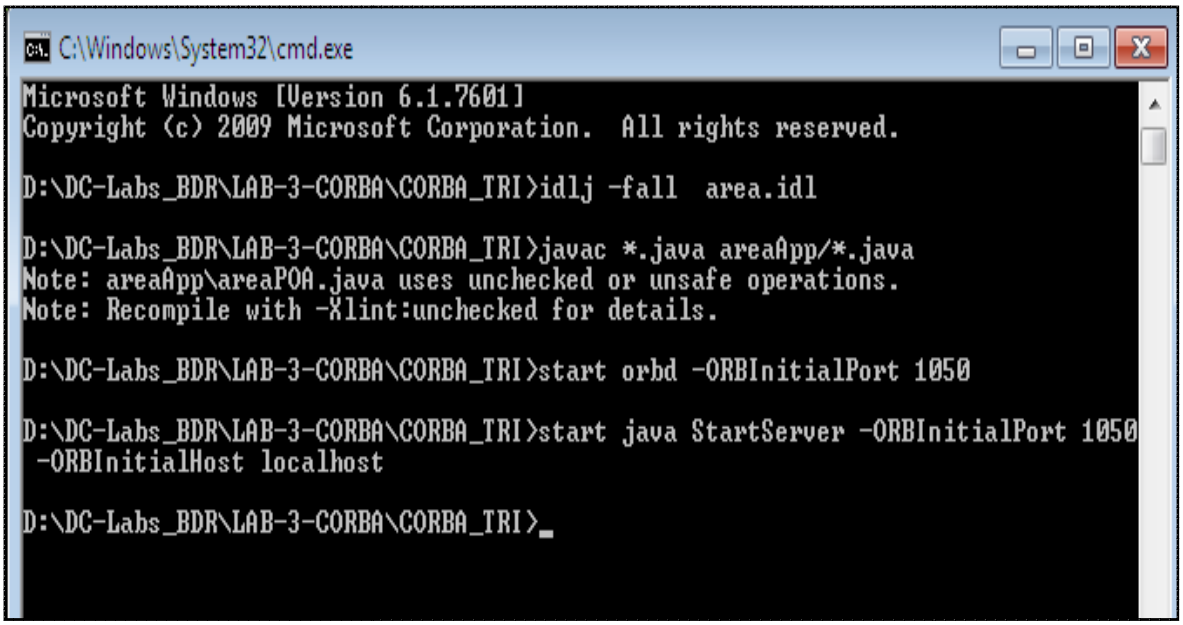
```
    /**
    * @param args the command line arguments
    */
   public static void main(String[] args) {
     try {
// create and initialize the ORB
            ORB orb = ORB.init(args, null);
// get the root naming context
            org.omg.CORBA.Object objRef =
orb.resolve_initial_references("NameService");
// Use NamingContextExt instead of NamingContext. This is
// part of the Interoperable naming Service.
            NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef);
// resolve the Object Reference in Naming
           area triobj = (area) areaHelper.narrow(ncRef.resolve_str("ABC"));
           Scanner c=new Scanner(System.in);
          System.out.println("Welcome to the circum system:");
                    System.out.println("Enter Height:");
                    String H = c.nextLine();
                    System.out.println("Enter Base:");
                    String B = c.nextLine();
                    double Height=Integer.parseInt(H);
                    double Base=Integer.parseInt(B);
                    double ans=triobj.area_tri(Height,Base);
                    System.out.println("The area of triangle is : "+ans);
                    System.out.println("-----------------------------------");
                }
      catch (Exception e) {
         System.out.println("Client exception: " + e);
           e.printStackTrace();
      }
    }

}
```

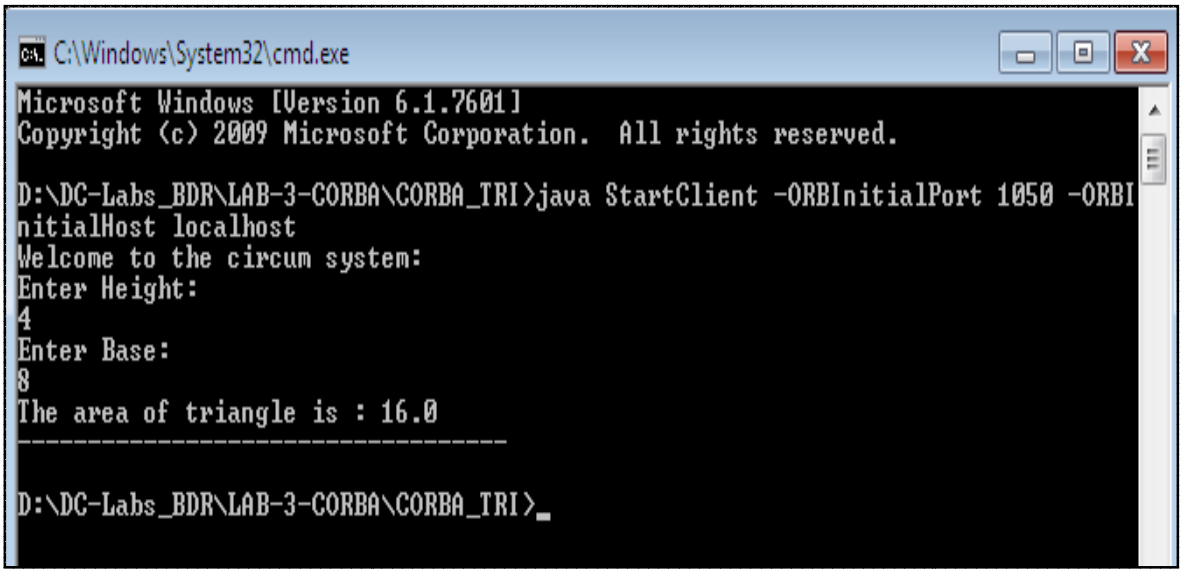
StartServer.java

```
import areaApp.*;
import org.omg.CosNaming.*;
import org.omg.CosNaming.NamingContextPackage.*;
import org.omg.CORBA.*;
import org.omg.PortableServer.*;
import org.omg.PortableServer.POA;
import java.util.Properties;
public class StartServer {

 public static void main(String args[]) {
   try{
// create and initialize the ORB
     ORB orb = ORB.init(args, null);
// get reference to rootpoa &amp; activate the POAManager
     POA rootpoa =
POAHelper.narrow(orb.resolve_initial_references("RootPOA"));
     rootpoa.the_POAManager().activate();

// create servant and register it with the ORB
     areaObj triobj = new areaObj();
     triobj.setORB(orb);

// get object reference from the servant
     org.omg.CORBA.Object ref = rootpoa.servant_to_reference(triobj);
     area href = areaHelper.narrow(ref);
//Gets the root naming context
     org.omg.CORBA.Object objRef =
orb.resolve_initial_references("NameService");
// Use NamingContextExt which is part of the Interoperable
// Naming Service (INS) specification.
     NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef);
// bind the Object Reference in Naming
     NameComponent path[] = ncRef.to_name( "ABC" );
```

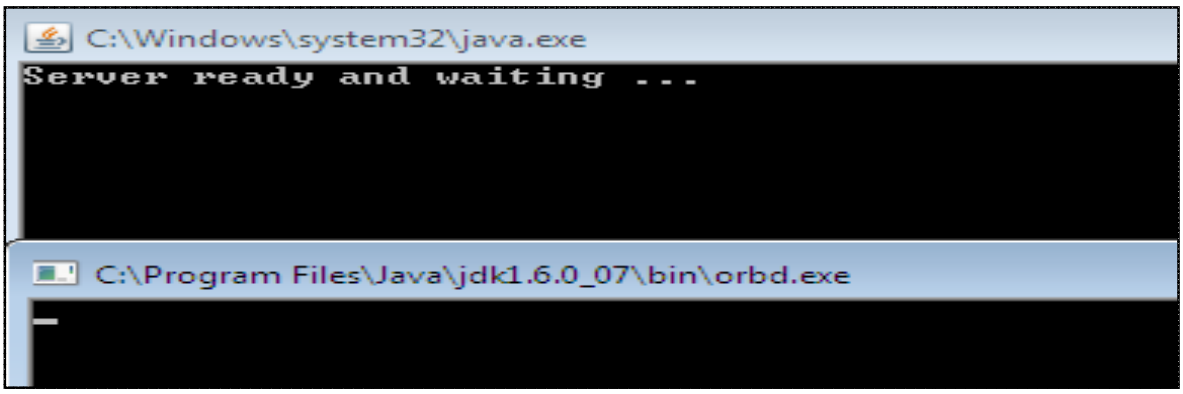
```
    ncRef.rebind(path, href);
    System.out.println("Server ready and waiting ...");
    // wait for invocations from clients
        orb.run();
      }
    catch (Exception e) {
      System.err.println("ERROR: " + e);
      e.printStackTrace(System.out);
    }
   System.out.println("Server Exiting ...");
  }
}
```

OUTPUT:




C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
D:\DC-Labs_BDR\LAB-3-CORBA\CORBA_TRI>java StartClient -ORBInitialPort 1050 -ORBInitialHost localhost
Welcome to the circum system:
Enter Height:
4
Enter Base:
8
The area of triangle is : 16.0

D:\DC-Labs_BDR\LAB-3-CORBA\CORBA_TRI>


C:\Windows\system32\java.exe
Server ready and waiting ...
C:\Program Files\Java\jdk1.6.0_07\bin\orbd.exe

LAB-4

AIM: Implement the program for Message Passing Interface.

Hello.c : to print hello world message

```
#include <stdio.h>
#include <mpi.h>
main(int argc, char **argv)
{
   int node;
   MPI_Init(&argc,&argv);
   MPI_Comm_rank(MPI_COMM_WORLD, &node);
  printf("Hello World from Node %d\n",node);            MPI_Finalize();
}
```

Host_file

```
localhost
localhost
localhost
localhost
```

Host file contains the addresses of the machines on which we want to pass the message.

Compile the program:

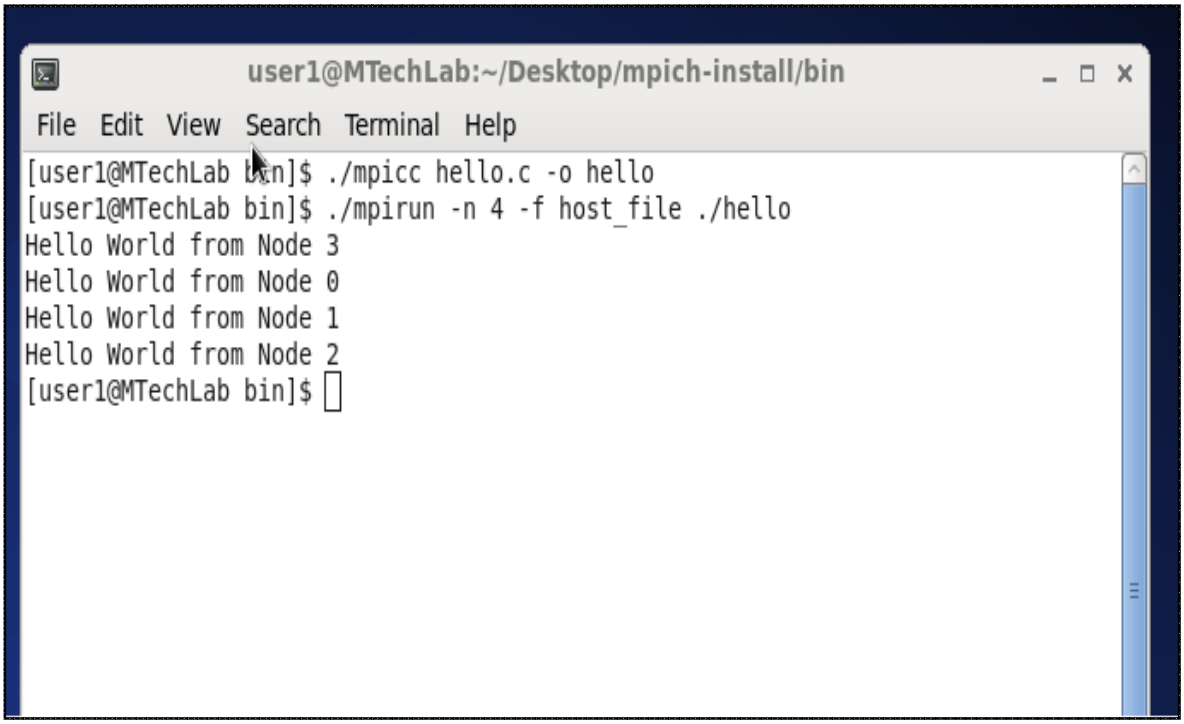
```
$ ./mpicc Hello.c -o hello
```

Execute the program

```
./mpirun -n 4 -f Host_file ./hello
```

OUTPUT:

Hello World from Node 0
Hello World from Node 1
Hello World from Node 2
Hello World from Node 3



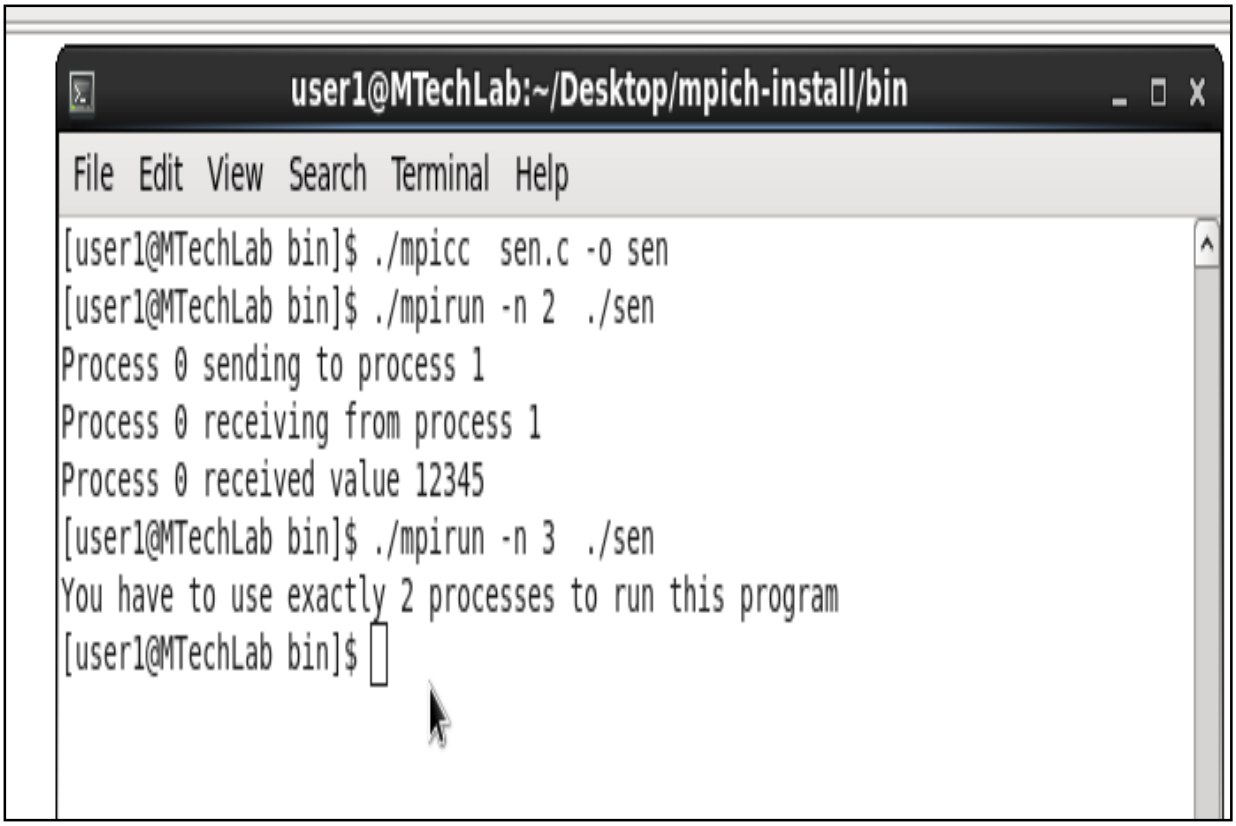
sen.c

/*A simple MPI example program using standard mode send and receive. The program consists of two processes. Process 0 sends a message to the receiver. This receives the message and sends it back.Compile the program with 'mpicc -O3 send-standard.c -o send-standard‘ Run it on two processes. */

```
#include <stdlib.h>
#include <stdio.h>
#include "mpi.h"
int main(int argc, char* argv[]) {
  int x, y, np, me;
  int tag = 42;
  MPI_Status status;
  /* Initialize MPI */
  MPI_Init(&argc, &argv);
MPI_Comm_size(MPI_COMM_WORLD, &np);
  MPI_Comm_rank(MPI_COMM_WORLD, &me);

/* Check that we run on exactly two proceses */
  if (np != 2) {
    if (me == 0) {
   printf("You have to use exactly 2 processes to run this program\n");
    }
    MPI_Finalize();      /* Quit if there is only one process */
    exit(0);
  }
x = 12345; y = me;   /* Initialize */
  if (me == 0) {
    printf("Process %d sending to process 1\n", me);
    MPI_Send(&x, 1, MPI_INT, 1, tag, MPI_COMM_WORLD);
    printf("Process %d receiving from process 1\n", me);
    MPI_Recv (&y, 1, MPI_INT, 1, tag, MPI_COMM_WORLD, &status);
    printf ("Process %d received value %d\n", me, y);
```

```
  } else { /* me == 1 */

    MPI_Recv (&y, 1, MPI_INT, 0, tag, MPI_COMM_WORLD, &status);
    MPI_Send (&y, 1, MPI_INT, 0, tag, MPI_COMM_WORLD);
  }

  MPI_Finalize();
  exit(0);
}
```

OUTPUT:



LAB-5

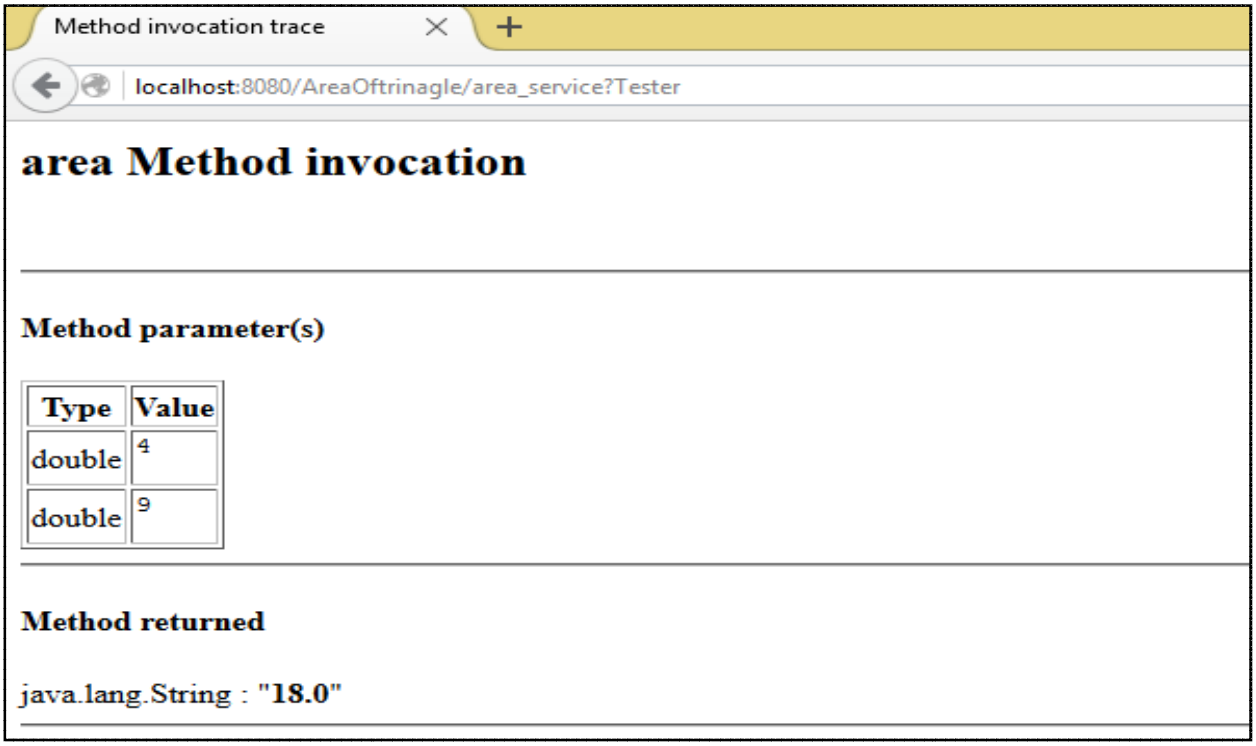
AIM: Implement the program for Web services. (to calculate area of triangle)

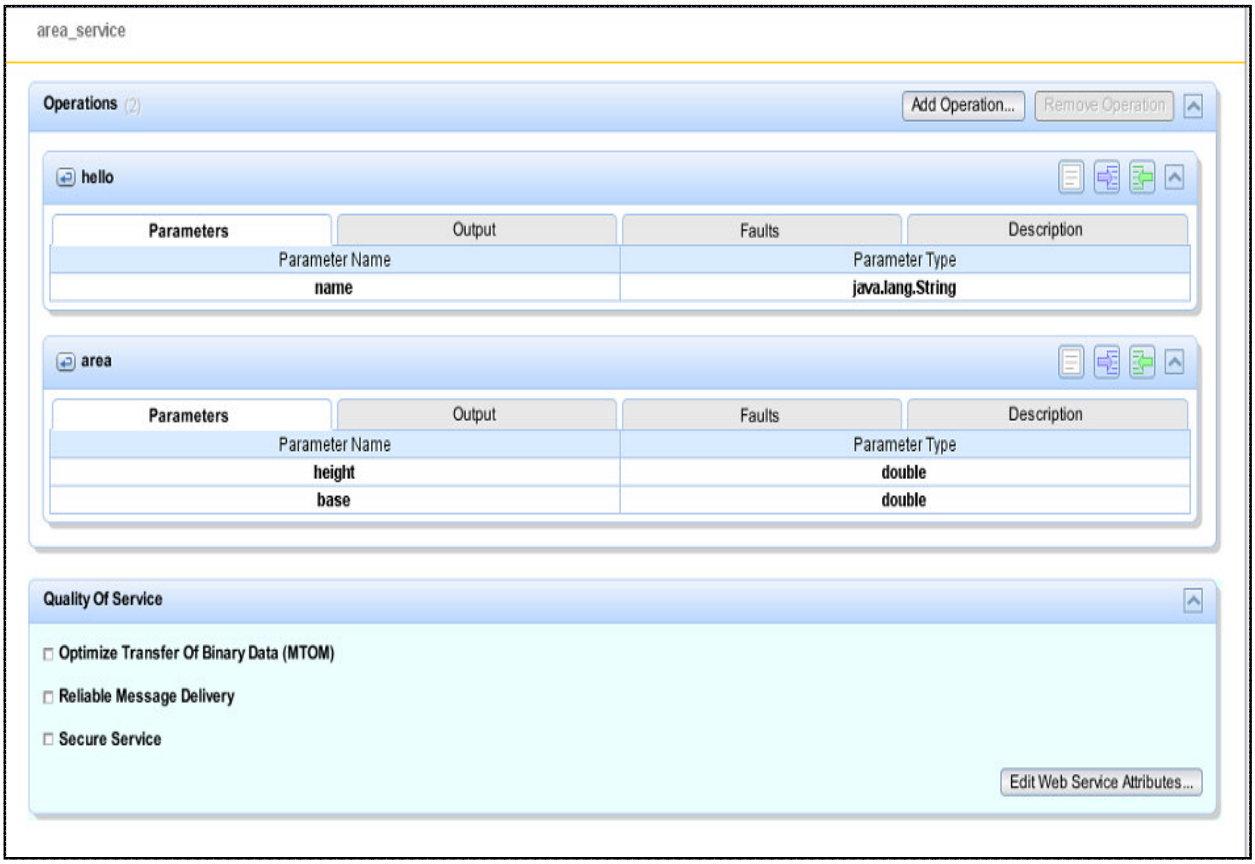
area_service.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package myp;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

@WebService(serviceName = "area_service")
public class area_service {

    /**
     * This is a sample web service operation
     */
    @WebMethod(operationName = "hello")
    public String hello(@WebParam(name = "name") String txt) {
        return "Hello " + txt + " !";
    }
    /**
     * Web service operation
     */
    @WebMethod(operationName = "area")
    public String area(@WebParam(name = "height") double height,
@WebParam(name = "base") double base) {
        //TODO write your implementation code here:
        return (0.5*height*base)+"";
```

```
        }
}
```

OUTPUT:


Method invocation trace
localhost:8080/AreaOftrinagle/area_service?Tester
area Method invocation
Method parameter(s)
Type
Value
double
4
double
9
Method returned
java.lang.String : "18.0"



ConsoleApp.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package consoleapp;
import java.util.*;
public class ConsoleAPP {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
```

```
        Scanner scan = new Scanner(System.in);
         System.out.print("Enter height : ");
        double h = scan.nextDouble();

        System.out.print("Enter base : ");
        double b = scan.nextDouble();
        System.out.println(area(h,b));
    }

    private static String area(double height, double base) {
        myp.AreaService_Service service = new myp.AreaService_Service();
        myp.AreaService port = service.getAreaServicePort();
        return port.area(height, base);
    }
}
```

OUTPUT:

```
Java DB Database Process X | GlassFish Server 4.0 X | Retriever Output X | ConsoleAPP (run) X | ConsoleAPP (run) #2 X

ant -f C:\\Users\\Joy\\Documents\\NetBeansProjects\\ConsoleAPP -Dnb.internal.action.name=run run
init:
Deleting: C:\Users\Joy\Documents\NetBeansProjects\ConsoleAPP\build\built-jar.properties
deps-jar:
Updating property file: C:\Users\Joy\Documents\NetBeansProjects\ConsoleAPP\build\built-jar.properties
wsimport-init:
wsimport-client-area_service:
files are up to date
wsimport-client-area_service_1:
files are up to date
wsimport-client-generate:
compile:
run:
Enter height : 4
Enter base : 8
16.0
BUILD SUCCESSFUL (total time: 8 seconds)
```

LABWORK BEYOND CURRICULA

Experiment 1

AIM: Mobile Agent (IBM's Aglet) Programming

What is Mobile Agent?

- Mobile agents are a distributed computing paradigm
- A mobile agent migrates from one machine to another under its own control
- Decides which locations to visit and what commands to execute
- Continuous interaction with the original source is not required
- Suspend execution at any point in time, transport itself to a new machine and resume execution

Toolkits: Mobile Agent Toolkits

Provide the infrastructure for mobile agents ...

– to interact with a local computer system; this is known as a "context" for mobile agents to reside and execute

– to move from host to host

– to communicate with other agents and hosts through message passing

– to maintain privacy and integrity (of agents as well as hosts)

These toolkits are normally Java-based e.g.

– Aglets

– Concordia

– JADE

– OAA

– TACOMA (C++)

What is Aglets ?

Java based mobile agent toolkit developed by IBM

– The name originates from Aglet =Agent + Applet

Download and Install

(1) Java 1.1.8_010/_16 JDK - http://java.sun.com/products/archive/index.html

(2) Aglet SDK- http://www.trl.ibm.com/aglets/idoagree103.htm (http://www.trl.ibm.co.jp/aglets)

Experiment 2

AIM: Implement Network File System (NFS)
Toolkits: Implementation of Clustering using MPI_CH2.
Steps:
Set up Network File System (NFS)
Set up Secure Shell (SSH)
Set up Message Passing Interface (MPI)

Requirement: 2 machine running Linux
Set up Network File System

Consider we have two host machine with ip 10.10.3.4 and 10.10.3.3;
Now I want to make 10.10.3.4 as server and rest as client then to implement NFS file system between this two follow the following steps:

Step 1: Host with ip 10.10.3.4 edit following file as given
in /etc/exports put entry /home 10.10.3.3(rw, no_root_squash) //we want share home directory to client.

in /etc/host.deny put entry portmap:ALl
in /etc/hosts.allow put entry
Portmap: 10.10.3.3
lockd: : 10.10.3.3
rquotad : 10.10.3.3
mound: 10.10.3.3
statd : 10.10.3.3

Step 2: Execute following set of command from root on both machine to start the daemons.
rpc.portmap
rpc.mountd, rpc.nfsd
rpc.statd, rpc.lockd
rpc.rquotad

Step 3: Client machine execute following command to mount server directory on client machine
(If possible make firewall off).
mount 10.10.3.4:/home /mnt/newhome
where 10.10.3.4->server host
/mnt/newhome----directory on client to which server directory /home will be mounted