Assignment 1

AddClient.java

```java
import java.rmi.*;
public class AddClient {
public static void main(String args[]) {
try {
String addServerURL = "rmi://" + args[0] + "/AddServer";
AddServerIntf addServerIntf =
(AddServerIntf)Naming.lookup(addServerURL);
System.out.println("The first number is: " + args[1]);
double d1 = Double.valueOf(args[1]).doubleValue();
System.out.println("The second number is: " + args[2]);
double d2 = Double.valueOf(args[2]).doubleValue();
System.out.println("The sum is: " + addServerIntf.add(d1, d2));
}
catch(Exception e) {
System.out.println("Exception: " + e);
}
}
}
```

AddServer.java

```java
import java.net.*;
import java.rmi.*;
public class AddServer {
public static void main(String args[]) {
try {
AddServerImpl addServerImpl = new AddServerImpl();
Naming.rebind("AddServer", addServerImpl);
}
catch(Exception e) {
System.out.println("Exception: " + e);
}
}
}
```
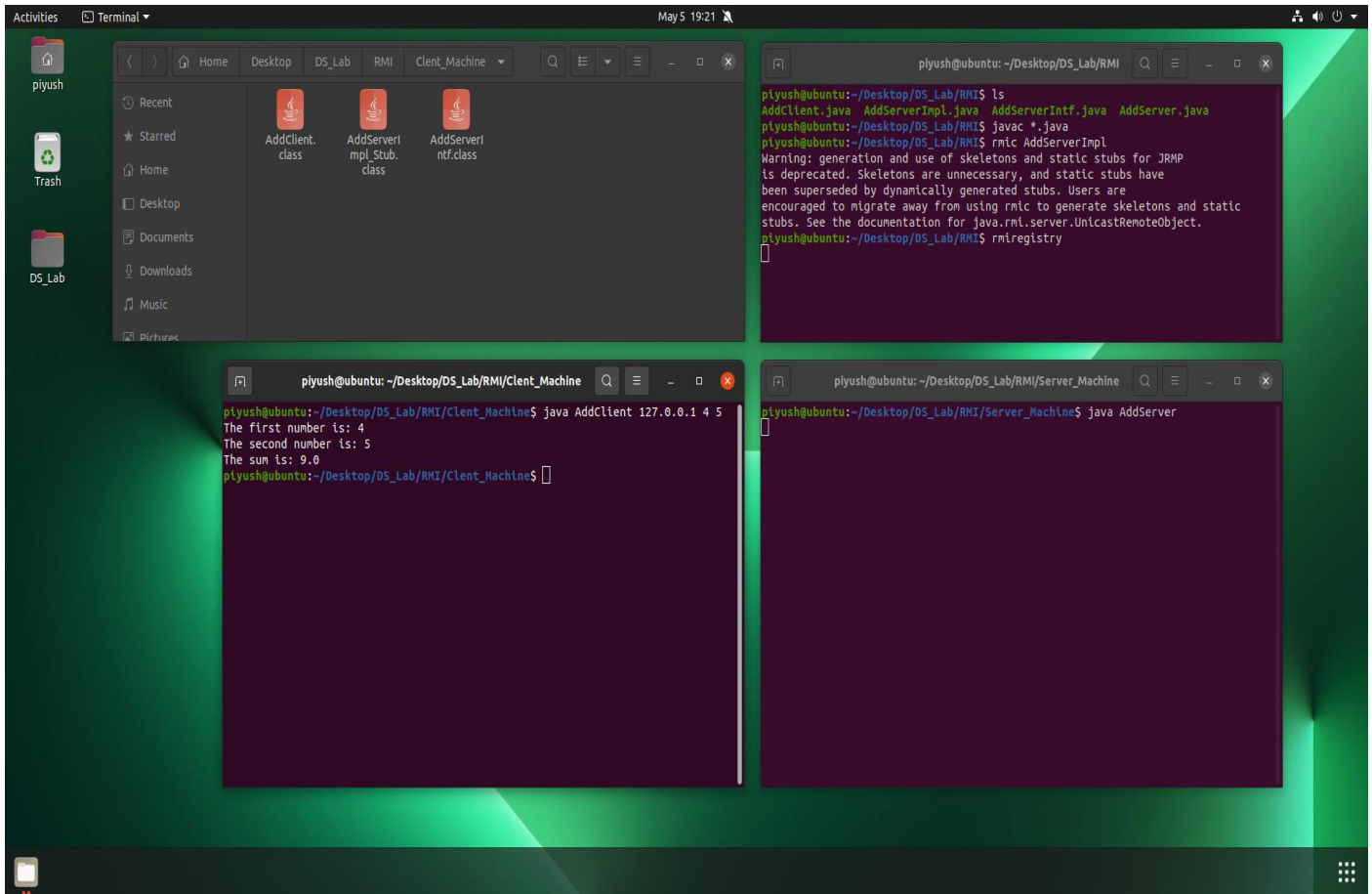
AddServerImpl.java

```java
import java.rmi.*;
import java.rmi.server.*;
public class AddServerImpl extends UnicastRemoteObject
implements AddServerIntf {
public AddServerImpl() throws RemoteException {
}
public double add(double d1, double d2) throws RemoteException {
return d1 + d2;
}
}
```

AddServerIntf.java

```java
import java.rmi.*;
public interface AddServerIntf extends Remote {
double add(double d1, double d2) throws RemoteException;
}
```

OUTPUT :

Assignment 2

ReverseClient.java

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

class ReverseClient
{
    public static void main(String args[])
    {
        Reverse ReverseImpl=null;
        try
        {
            org.omg.CORBA.ORB orb = org.omg.CORBA.ORB.init(args,null);

            org.omg.CORBA.Object objRef = orb.resolve_initial_references("NameService");
            NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef);

            String name = "Reverse";
            ReverseImpl = ReverseHelper.narrow(ncRef.resolve_str(name));

            System.out.println("Enter String=");
            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
            String str= br.readLine();

            String tempStr= ReverseImpl.reverse_string(str);
            System.out.println(tempStr);
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

ReverseImpl.java

```java
import ReverseModule.ReversePOA;
import java.lang.String;
class ReverseImpl extends ReversePOA
{
    ReverseImpl()
    {
        super();
        System.out.println("Reverse Object Created");
    }
    public String reverse_string(String name)
    {
    StringBuffer str=new StringBuffer(name);
    str.reverse();
        return (("Server Send "+str));
    }
}
```

ReverseServer.java

```java
import ReverseModule.Reverse;
import org.omg.CosNaming.*;
import org.omg.CosNaming.NamingContextPackage.*;
import org.omg.CORBA.*;
import org.omg.PortableServer.*;

class ReverseServer
{
    public static void main(String[] args)
    {
        try
        {
            // initialize the ORB
            org.omg.CORBA.ORB orb = org.omg.CORBA.ORB.init(args,null);

            // initialize the BOA/POA
            POA rootPOA = POAHelper.narrow(orb.resolve_initial_references("RootPOA"));
            rootPOA.the_POAManager().activate();

            // creating the  object
            ReverseImpl rvr = new ReverseImpl();

            // get the object reference from the servant class
            org.omg.CORBA.Object ref = rootPOA.servant_to_reference(rvr);

            System.out.println("Step1");
            Reverse h_ref = ReverseModule.ReverseHelper.narrow(ref);
            System.out.println("Step2");

            org.omg.CORBA.Object objRef = orb.resolve_initial_references("NameService");

            System.out.println("Step3");
            NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef);
            System.out.println("Step4");

            String name = "Reverse";
            NameComponent path[] = ncRef.to_name(name);
            ncRef.rebind(path,h_ref);

            System.out.println("Reverse Server reading and waiting....");
            orb.run();
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
```
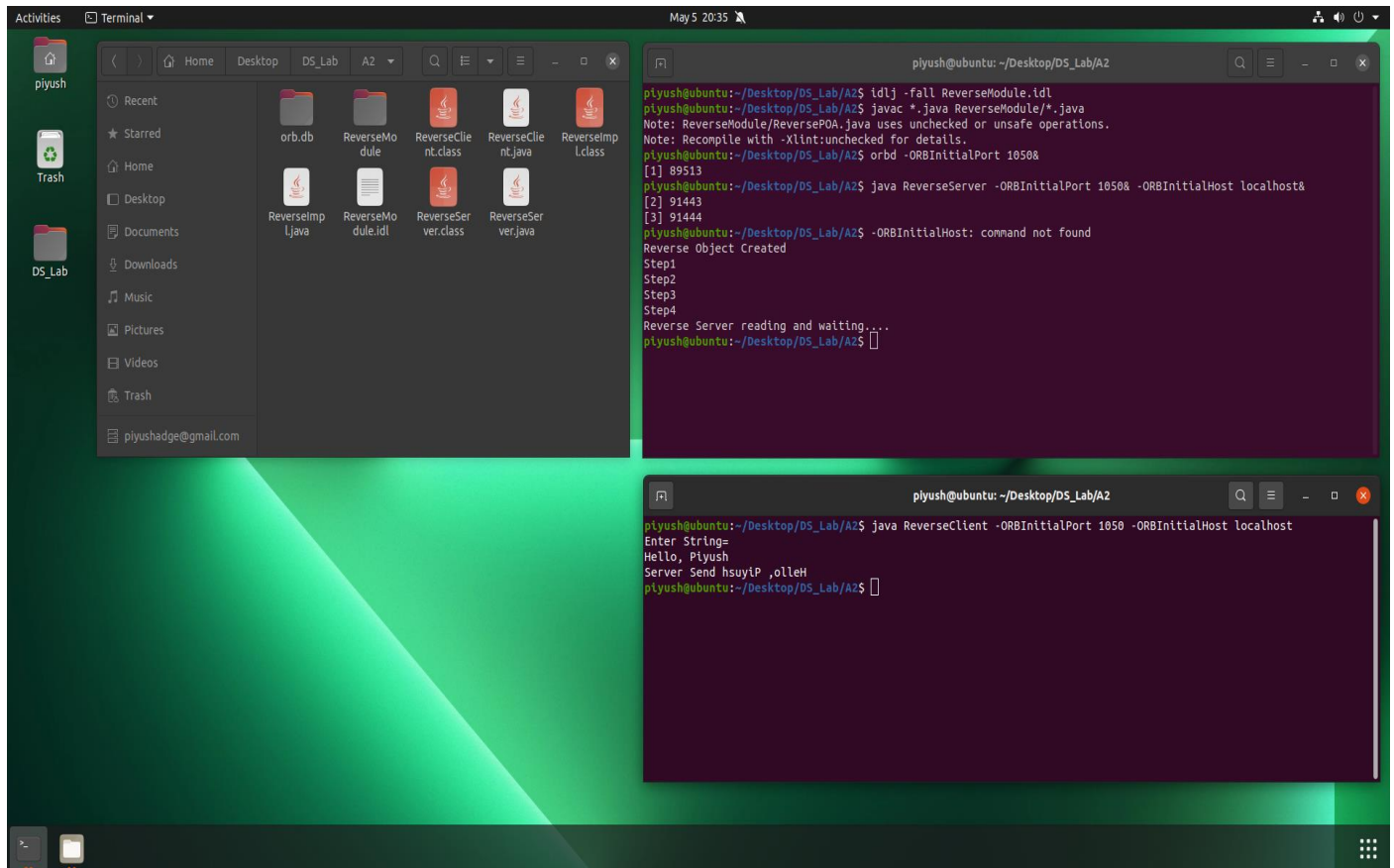
OUTPUT :

Assignment 3

CODE :

```c
#include <stdio.h>
#include <mpi.h>
int main(int argc, char** argv) {
    int rank, size;
    int arr[10] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
    int sum = 0;
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);
    int n = sizeof(arr) / sizeof(arr[0]);
    int local_sum = 0;
    for (int i = rank; i < n; i += size) {
        local_sum += arr[i];
    }
    MPI_Reduce(&local_sum, &sum, 1, MPI_INT, MPI_SUM, 0, MPI_COMM_WORLD);
    if (rank == 0) {
        printf("Sum of array elements is %d\n", sum);
    }
    MPI_Finalize();
}
```
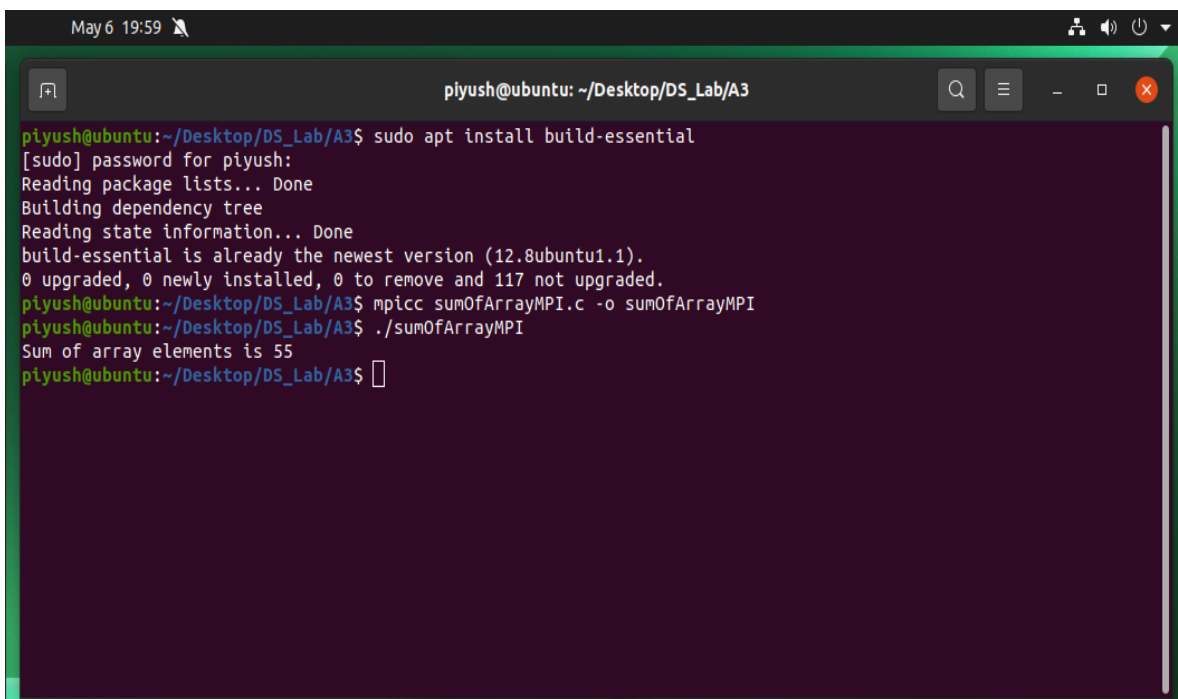
OUTPUT :

Code :

```java
import java.io.*;
import java.net.*;
import java.util.*;
public class BerkeleyAlgorithm {
    private static final int PORT = 1024;
    public static void main(String[] args) throws Exception {
        ServerSocket serverSocket = new ServerSocket(PORT);
        List<Long> timeDiffs = new ArrayList<Long>();
        Thread timeServerThread = new Thread(new Runnable() {
            public void run() {
                while (true) {
                    try {
                        Socket clientSocket = serverSocket.accept();
                        ObjectInputStream in = new ObjectInputStream(clientSocket.getInputStream());
                        Date clientTime = (Date) in.readObject();
                        ObjectOutputStream out = new ObjectOutputStream(clientSocket.getOutputStream());
                        out.writeObject(new Date());
                        long timeDiff = (new Date().getTime() - clientTime.getTime()) / 2;
                        timeDiffs.add(timeDiff);
                        in.close();
                        out.close();
                        clientSocket.close();
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                }
            }
        });
        timeServerThread.start();

        Thread timeClientThread = new Thread(new Runnable() {
            public void run() {
                while (true) {
```

```java
                try {
                    Socket socket = new Socket("localhost", PORT);
                    ObjectOutputStream out = new ObjectOutputStream(socket.getOutputStream());
                    out.writeObject(new Date());
                    ObjectInputStream in = new ObjectInputStream(socket.getInputStream());
                    Date serverTime = (Date) in.readObject();
                    long timeDiff = (serverTime.getTime() - new Date().getTime()) / 2;
                    timeDiffs.add(timeDiff);
                    in.close();
                    out.close();
                    socket.close();
                    Thread.sleep(1000);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        }
    });
    timeClientThread.start();
    Thread.sleep(10000);
    long sumTimeDiff = 0;
    for (Long timeDiff : timeDiffs) {
        sumTimeDiff += timeDiff;
    }
    long avgTimeDiff = sumTimeDiff / timeDiffs.size();
    System.out.println("Average time difference: " + avgTimeDiff);
    Calendar calendar = Calendar.getInstance();
    calendar.setTime(new Date());
    calendar.add(Calendar.MILLISECOND, (int) avgTimeDiff);
    System.out.println("Adjusted time: " + calendar.getTime());
}
}
```

OUTPUT :

CODE :

```java
import java.io.*;
import java.util.*;

class Tok {
    public static void main(String args[]) throws Throwable {
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter the num of nodes:");
        int n = scan.nextInt();
        int m = n - 1;
        int token = 0;
        int ch = 0, flag = 0;
        for (int i = 0; i < n; i++) {
            System.out.print(" " + i);
        }
        System.out.println(" " + 0);
        do{
            System.out.println("Enter sender:");
            int s = scan.nextInt();
            System.out.println("Enter receiver:");
            int r = scan.nextInt();
            System.out.println("Enter Data:");
            int a;
            a = scan.nextInt();
            System.out.print("Token passing:");
            for (int i = token, j = token; (i % n) != s; i++, j = (j + 1) % n) {
                System.out.print(" " + j + "->");
            }
            System.out.println(" " + s);
            System.out.println("Sender " + s + " sending data: " + a);
            for (int i = s + 1; i != r; i = (i + 1) % n) {
                System.out.println("data  " + a + " forwarded by " + i);
            }
```

```java
            System.out.println("Receiver " + r + " received data: " + a +"\n");
            token = s;
            do{
               try {
                  if( flag == 1)
                           System.out.print("Invalid Input!!...");
                  System.out.print("Do you want to send again?? enter 1 for Yes and 0 for No : ");
                  ch = scan.nextInt();
                  if( ch != 1 && ch != 0 )
                           flag = 1;
                  else
                           flag = 0;
               } catch (InputMismatchException e){
                  System.out.println("Invalid Input");
               }
            }while( ch != 1 && ch != 0 );
         }while( ch == 1 );
      }
}
```

OUTPUT :

```
Problems  Javadoc  Declaration  Console ×
<terminated> Tok [Java Application] /snap/eclipse/66/plugins/org.eclipse.justj.openjdk.hotspot.jre.f
Enter the num of nodes:
4
 0 1 2 3 0
Enter sender:
1
Enter receiver:
3
Enter Data:
15
Token passing: 0-> 1
Sender 1 sending data: 15
data  15 forwarded by 2
Receiver 3 received data: 15

Do you want to send again?? enter 1 for Yes and 0 for No : 0
```

Bully Algorithm :

```java
import java.io.InputStream;
import java.io.PrintStream;
import java.util.Scanner;
public class Bully {
    static boolean[] state = new boolean[5];
    int coordinator;
    public static void up(int up) {
        if (state[up - 1]) {
            System.out.println("process" + up + "is already up");
        } else {
            int i;
            Bully.state[up - 1] = true;
            System.out.println("process " + up + "held election");
            for (i = up; i < 5; ++i) {
                System.out.println("election message sent from process" + up + "to process" + (i + 1));
            }
            for (i = up + 1; i <= 5; ++i) {
                if (!state[i - 1]) continue;
                System.out.println("alive message send from process" + i + "to process" + up);
                break;
            }
        }
    }
    public static void down(int down) {
        if (!state[down - 1]) {
            System.out.println("process " + down + "is already dowm.");
        } else {
            Bully.state[down - 1] = false;
        }
    }
    public static void mess(int mess) {
        if (state[mess - 1]) {
            if (state[4]) {
```

```java
                System.out.println("OK");
            } else if (!state[4]) {
                int i;
                System.out.println("process" + mess + "election");
                for (i = mess; i < 5; ++i) {
                    System.out.println("election send from process" + mess + "to process " + (i + 1));
                }
                for (i = 5; i >= mess; --i) {
                    if (!state[i - 1]) continue;
                    System.out.println("Coordinator message send from process" + i + "to all");
                    break;
                }
            }
        } else {
            System.out.println("Prccess" + mess + "is down");
        }
    }
    public static void main(String[] args) {
        int choice;
        Scanner sc = new Scanner(System.in);
        for (int i = 0; i < 5; ++i) {
            Bully.state[i] = true;
        }
        System.out.println("5 active process are:");
        System.out.println("Process up  = p1 p2 p3 p4 p5");
        System.out.println("Process 5 is coordinator");
        do {
            System.out.println(".........");
            System.out.println("1 up a process.");
            System.out.println("2.down a process");
            System.out.println("3 send a message");
            System.out.println("4.Exit");
            choice = sc.nextInt();
            switch (choice) {
                case 1: {
```

```java
                System.out.println("bring proces up");

                int up = sc.nextInt();

                if (up == 5) {

                    System.out.println("process 5 is co-ordinator");

                    Bully.state[4] = true;

                    break;

                }

                Bully.up(up);

                break;

            }

            case 2: {

                System.out.println("bring down any process.");

                int down = sc.nextInt();

                Bully.down(down);

                break;

            }

            case 3: {

                System.out.println("which process will send message");

                int mess = sc.nextInt();

                Bully.mess(mess);

            }

        }

    } while (choice != 4);

    }

}
```

Ring Algorithm :

```java
import java.util.Scanner;

public class Ring {

    public static void main(String[] args) {

        int temp, i, j;
        char str[] = new char[10];
        Rr proc[] = new Rr[10];

        for (i = 0; i < proc.length; i++)
```

```java
            proc[i] = new Rr();

    Scanner in = new Scanner(System.in);
    System.out.println("Enter the number of process : ");
    int num = in.nextInt();

    for (i = 0; i < num; i++) {
            proc[i].index = i;
            System.out.println("Enter the id of process : ");
            proc[i].id = in.nextInt();
            proc[i].state = "active";
            proc[i].f = 0;
    }

    for (i = 0; i < num - 1; i++) {
            for (j = 0; j < num - 1; j++) {
                    if (proc[j].id > proc[j + 1].id) {
                            temp = proc[j].id;
                            proc[j].id = proc[j + 1].id;
                            proc[j + 1].id = temp;
                    }
            }
    }

    for (i = 0; i < num; i++) {
            System.out.print("  [" + i + "]" + " " + proc[i].id);
    }

    int init;
    int ch;
    int temp1;
    int temp2;
    int ch1;
    int arr[] = new int[10];

    proc[num - 1].state = "inactive";

    System.out.println("\n process " + proc[num - 1].id + " select as co-ordinator");

    while (true) {
            System.out.println("\n 1.election 2.quit ");
            ch = in.nextInt();

            for (i = 0; i < num; i++) {
                    proc[i].f = 0;
            }

            switch (ch) {
            case 1:
                    System.out.println("\n Enter the Process number who initialsied election : ");
                    init = in.nextInt();
                    temp2 = init;
                    temp1 = init + 1;

                    i = 0;
```

```java
                                while (temp2 != temp1) {
                                        if ("active".equals(proc[temp1].state) && proc[temp1].f == 0) {

                                                System.out.println("\nProcess " + proc[init].id + " send
message to " + proc[temp1].id);

                                                proc[temp1].f = 1;
                                                init = temp1;
                                                arr[i] = proc[temp1].id;
                                                i++;

                                        }
                                        if (temp1 == num) {
                                                temp1 = 0;
                                        } else {
                                                temp1++;
                                        }
                                }

                                System.out.println("\nProcess " + proc[init].id + " send message to " +
proc[temp1].id);

                                arr[i] = proc[temp1].id;
                                i++;
                                int max = -1;

                                for (j = 0; j < i; j++) {
                                        if (max < arr[j]) {
                                                max = arr[j];
                                        }
                                }
                                System.out.println("\n process " + max + "select as co-ordinator");

                                for (i = 0; i < num; i++) {

                                        if (proc[i].id == max) {
                                                proc[i].state = "inactive";
                                        }
                                }
                                break;
                        case 2:
        System.out.println("Program terminated ...");
        return ;
                        default:
                                System.out.println("\n invalid response \n");
                                break;
                        }
                }
        }
}

class Rr {

        public int index;
        public int id;
        public int f;
        String state;
```

OUTPUT :

```
Console ×
Bully [Java Application] C:\Users\sanke\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\javaw.exe (04-May-2023, 11:20:25
5 active process are:
Process up  = p1 p2 p3 p4 p5
Process 5 is coordinator
.........
1 up a process.
2.down a process
3 send a message
4.Exit
2
bring down any process.
5
.........
1 up a process.
2.down a process
3 send a message
4.Exit
3
which process will send message
2
process2election
election send from process2to process 3
election send from process2to process 4
election send from process2to process 5
Coordinator message send from process4to all
.........
1 up a process.
2.down a process
3 send a message
4.Exit
```

```
Problems  @ Javadoc  Declaration  Console ×
Ring [Java Application] C:\Users\sanke\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\javaw.exe (04-May-2023, 11:06:59
Enter the number of process :
4
Enter the id of process :
1
Enter the id of process :
2
Enter the id of process :
3
Enter the id of process :
4
  [0] 1  [1] 2  [2] 3  [3] 4
 process 4select as co-ordinator

 1.election 2.quit
1

 Enter the Process number who initialsied election :
2

Process 3 send message to 1

Process 1 send message to 2

Process 2 send message to 3

 process 3select as co-ordinator

 1.election 2.quit
```

Assignment 7

Web Services :

Addition.java

```java
package addition;

import javax.jws.WebService;

import javax.jws.WebMethod;

import javax.jws.WebParam;

@WebService(serviceName = "Addition")

public class Addition {

    /* Web service operation  */

    @WebMethod(operationName = "addition")

    public Integer addition(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {

        //TODO write your implementation code here:

        return a+b;

    }

}
```

Web Service References :

Index.jsp

```jsp
<%--

    Document   : index

    Created on : 7 May, 2023, 1:10:25 AM

    Author     : kiran

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

    <head>

        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

        <title>JSP Page</title>

    </head>

    <body>

        <form action="index.jsp">

            <input type="text" name="a">

            <br>
```
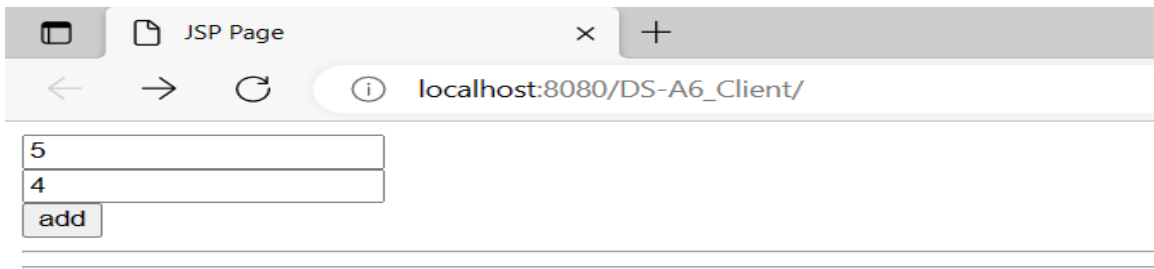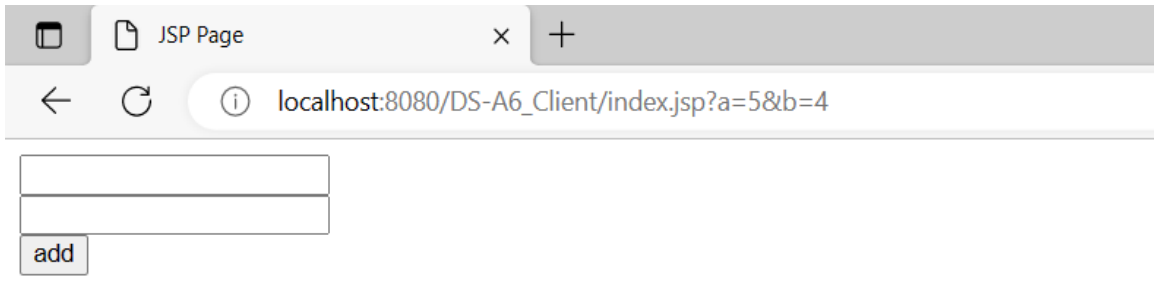
```
        <input type="text" name="b">
        <br>
        <input type="submit" value="add">
    </form>
    <%-- start web service invocation --%><hr/>
    <%
    try {
        addition.Addition_Service service = new addition.Addition_Service();
        addition.Addition port = service.getAdditionPort();
         // TODO initialize WS operation arguments here
        String a = request.getParameter("a");
        String b = request.getParameter("b");
      int aa = Integer.parseInt(a);
      int bb = Integer.parseInt(b);
        // TODO process result here
        java.lang.Integer result = port.addition(aa, bb);
        out.println("Result = "+result);
    } catch (Exception ex) {
        // TODO handle custom exceptions here
    }
    %>
    <%-- end web service invocation --%><hr/>
    </body>
</html>
```
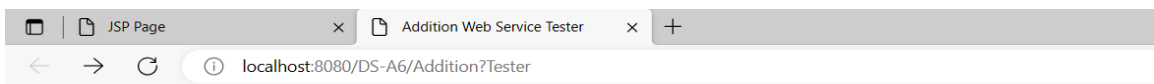
OUTPUT :

localhost:8080/DS-A6_Client/

```
5
4
add
```


localhost:8080/DS-A6_Client/index.jsp?a=5&b=4

```
add
```
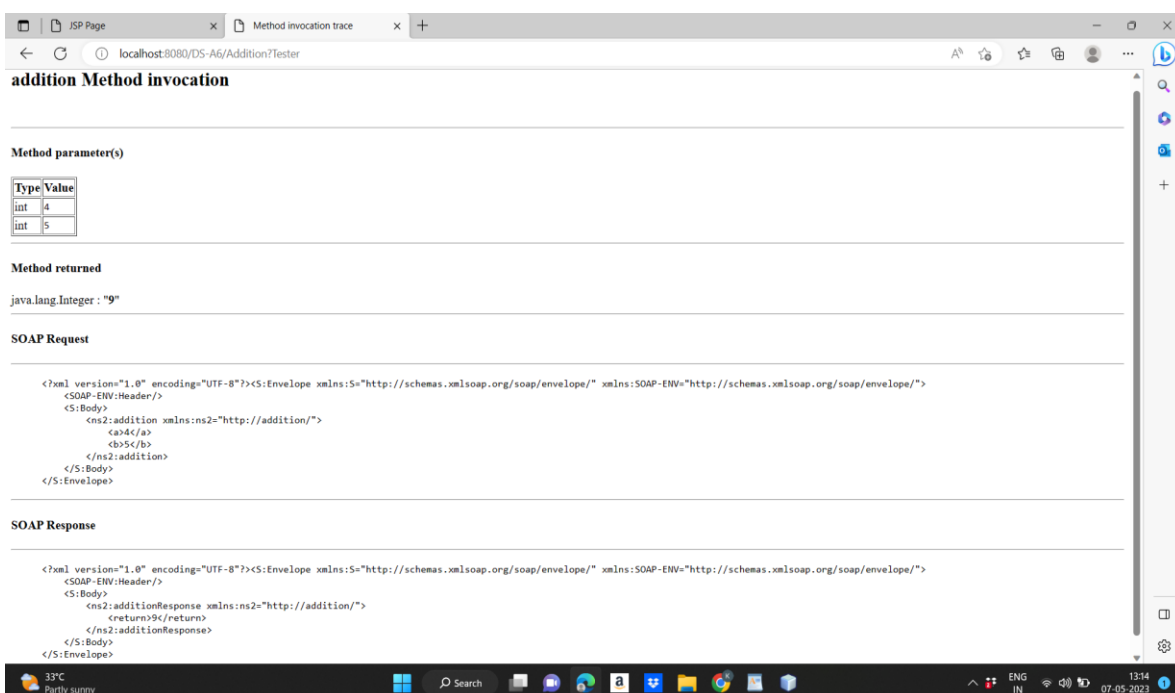
Result = 9

localhost:8080/DS-A6/Addition?Tester

Addition Web Service Tester

This form will allow you to test your web service implementation (WSDL File)

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

public abstract java.lang.Integer addition.Addition.addition(int,int)
addition (4 , 5)

localhost:8080/DS-A6/Addition?Tester

addition Method invocation

Method parameter(s)

Type	Value
int	4
int	5

Method returned

java.lang.Integer : "9"

SOAP Request

```xml
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:addition xmlns:ns2="http://addition/">
            <a>4</a>
            <b>5</b>
        </ns2:addition>
    </S:Body>
</S:Envelope>
```

SOAP Response

```xml
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:additionResponse xmlns:ns2="http://addition/">
            <return>9</return>
        </ns2:additionResponse>
    </S:Body>
</S:Envelope>
```