****************** Assignmemt-1***************

Title:To develop any distributed application through implementing client-server communication programs based on Java Sockets

*****************Client.java**************

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.InetAddress;
import java.net.Socket;

public class Client
{

    private static Socket socket;

    public static void main(String[] args) {

        try {
            String host = "localhost";
            int port = 25000;
            InetAddress add = InetAddress.getByName(host);
            socket = new Socket(add, port);

            //Send message to the server
            BufferedWriter bw = new BufferedWriter( new
OutputStreamWriter(socket.getOutputStream()));

            String number = "2";
            bw.write(number+"\n");
            bw.flush();
            System.out.println("Message sent to the server is " + number
+"\n");

            BufferedReader br = new BufferedReader( new
InputStreamReader(socket.getInputStream()));
            String msg = br.readLine();
            System.out.println("Message received from the server "+msg);
        }catch (Exception e) {
            e.printStackTrace();
        }finally {
            try {
                 socket.close();
            }catch (Exception e) {
                e.printStackTrace();
            }
        }
    }

}
```

```
*****************Server.java*****************

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.ServerSocket;
import java.net.Socket;

public class Server {

	private static Socket socket;

	public static void main(String[] args) {

		try {
			int port = 25000;
			ServerSocket serverSocket = new ServerSocket(port);
			System.out.println("Server started and listening");
			System.out.println("Client may send the number to
receive its double");
			while (true) {
				// Reading message from the client
				socket = serverSocket.accept();
				BufferedReader br = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
				String number = br.readLine();
				System.out.println("Message received from the
client "+ socket.getRemoteSocketAddress() +" is " + number);

				String returnMessage;
				try {
					int num = Integer.parseInt(number);
					int retval = num * 2;
					returnMessage = String.valueOf(retval) +
"\n";
				} catch (NumberFormatException e) {
					returnMessage = "Input incorrect. Please try
again";
				}

				BufferedWriter bw = new BufferedWriter(new
OutputStreamWriter(socket.getOutputStream()));
				bw.write(returnMessage);
				System.out.println("Message sent to the client is
" + returnMessage);
				bw.flush();
			}
		} catch (Exception e) {
			e.printStackTrace();
		} finally {
			try {
```

```
                    socket.close();
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        }
}
```

Output:

Serever:

```
Server started and listening
Client may send the number to receive its double
```

Client:

```
Message sent to the server is 2

Message received from the server 4
```

//******************Assignment 1[B]**************************
// Title: To develop any distributed application through implementing client-server communication programs based on RMI tchnique

```
*************** AddServer.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) {
                  e.printStackTrace();
            }
      }
}

**************** AddClient.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 first number is "+ args[2]);
                  double d2 = Double.valueOf(args[2]).doubleValue();
                  System.out.println("The sum is "+addServerIntf.add(d1,
d2));
            }catch (Exception e) {
                  e.printStackTrace();
            }
      }
}

****************** AddServerImpl.java ********************
import java.rmi.*;
import java.rmi.server.*;

public class AddServerImpl extends UnicastRemoteObject implements
AddServerIntf{

      protected AddServerImpl() throws RemoteException {
      }
      public double add(double d1, double d2) throws RemoteException{
            return d1+d2;
```

```
      }

}
```

```
****************** AddServerIntf.java *********************
import java.rmi.*;

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

Output

```
Terminal 1
gurukul@gurukul-ThinkCentre-M57e:~$ cd Desktop/dev/rmi/src
gurukul@gurukul-ThinkCentre-M57e:~/Desktop/dev/rmi/src$ javac *.java
gurukul@gurukul-ThinkCentre-M57e:~/Desktop/dev/rmi/src$ rmic
AddServerImpl
gurukul@gurukul-ThinkCentre-M57e:~/Desktop/dev/rmi/src$
----------------------------------------------------------------

Terminal 2
gurukul@gurukul-ThinkCentre-M57e:~$ cd Desktop/dev/rmi/src
gurukul@gurukul-ThinkCentre-M57e:~/Desktop/dev/rmi/src$ cd Server
gurukul@gurukul-ThinkCentre-M57e:~/Desktop/dev/rmi/src/Server$
rmiregistry
------------------------------------------------------------------------
Terminal 3

gurukul@gurukul-ThinkCentre-M57e:~/Desktop/dev/rmi/src/Server$ cd ..
gurukul@gurukul-ThinkCentre-M57e:~/Desktop/dev/rmi/src$ cd Client
gurukul@gurukul-ThinkCentre-M57e:~/Desktop/dev/rmi/src/Client$ java
AddClient 127.0.0.1 5 6
The first number is 5
The first number is 6
The sum is 11.0
gurukul@gurukul-ThinkCentre-M57e:~/Desktop/dev/rmi/src/Client$
```

```
//Assignment 2
Title:To develop any distributed application using Message Passing
Interface (MPI).


import mpi.MPI;

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

       MPI.Init(args);

       int rank = MPI.COMM_WORLD.Rank();

       int size = MPI.COMM_WORLD.Size();
       int root=0;

       int sendbuf[]=null;

       sendbuf= new int[size];


       if(rank==root){
             sendbuf[0] = 10;
             sendbuf[1] = 20;
             sendbuf[2] = 30;
             sendbuf[3] = 40;

             System.out.print("Processor "+rank+" has data: ");
             for(int i = 0; i < size; i++){
                   System.out.print(sendbuf[i]+" ");
             }
             System.out.println();
       }

       int recvbuf[] = new int[1];


       MPI.COMM_WORLD.Scatter(sendbuf, 0, 1, MPI.INT, recvbuf, 0, 1,
MPI.INT, root);
       System.out.println("Processor "+rank+" has data: "+recvbuf[0]);
       System.out.println("Processor "+rank+" is doubling the data");
       recvbuf[0]=recvbuf[0]*2;

       MPI.COMM_WORLD.Gather(recvbuf, 0, 1, MPI.INT, sendbuf, 0, 1,
MPI.INT, root);

       if(rank==root){
            System.out.println("Process 0 has data: ");
             for(int i=0;i<4;i++){
                   System.out.print(sendbuf[i]+ " ");
             }
```

```
        }

        MPI.Finalize();
    }
}
```

```
//********************** Output ********************
gurukul@gurukul-ThinkCentre-M57e:~$ export
MPJ_HOME=/home/gurukul/Desktop/MPI/mpj-v0_44
gurukul@gurukul-ThinkCentre-M57e:~$ cd Desktop
gurukul@gurukul-ThinkCentre-M57e:~/Desktop$ cd MPI
gurukul@gurukul-ThinkCentre-M57e:~/Desktop/MPI$ javac -cp
$MPJ_HOME/lib/mpj.jar mpj.java
gurukul@gurukul-ThinkCentre-M57e:~/Desktop/MPI$ $MPJ_HOME/bin/mpjrun.sh -
np 4 mpj
MPJ Express (0.44) is started in the multicore configuration
Processor 0 has data: 10 20 30 40
Processor 0 has data: 10
Processor 0 is doubling the data
Processor 1 has data: 20
Processor 1 is doubling the data
Processor 2 has data: 30
Processor 2 is doubling the data
Processor 3 has data: 40
Processor 3 is doubling the data
Process 0 has data:
20 40 60 80
gurukul@gurukul-ThinkCentre-M57e:~/Desktop/MPI$
```

Assignment 3

Title:To develop any distributed application with CORBA program using java IDL for reverse of string.

********** ReverseClient.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
        {
            // initialize the ORB object request broker
            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";

           // narrow converts generic object into string type
            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();
        }
    }
}
```

```
**********************  ReverseServer.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();
        }
    }
```

```
}
```

```
********************************** ReverseImpl.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));
    }
}

**************************************** ReverseModule.idl
****************************
module ReverseModule
{
    interface Reverse
    {
        string reverse_string(in string str);
    };
};
```

Output:
Terminal 1_Server:

```
gurukul@gurukul-ThinkCentre-M57e:~$ cd Desktop
gurukul@gurukul-ThinkCentre-M57e:~/Desktop$ ls
46 mpj                                          IDL CORBA
abc                                          mpj-v0_44.tar.gz
ahirers1dsfl                                    pai
alphabetized-cassette-tapes.zip                      pp.sh~
Annamarie-Farah-St-Bishoy-Coptic-Orthodox-College-NSW.jpg  pra.sh~
ass1.cpp~                                     stack.cpp~
ass1.sh~                                     TCP
Assignment 1_files                              Untitled Document~
Assignment 1.html                            vaishali11.txt
assignment-name(1).pdf                          vaishalimore
gurukul@gurukul-ThinkCentre-M57e:~/Desktop$ cd CORBA
gurukul@gurukul-ThinkCentre-M57e:~/Desktop/CORBA$ idlj -fall
ReverseModule.idl
gurukul@gurukul-ThinkCentre-M57e:~/Desktop/CORBA$ clear

gurukul@gurukul-ThinkCentre-M57e:~/Desktop/CORBA$ cd ..
gurukul@gurukul-ThinkCentre-M57e:~/Desktop$  cd..
cd..: command not found
gurukul@gurukul-ThinkCentre-M57e:~/Desktop$  cd ..
gurukul@gurukul-ThinkCentre-M57e:~$ cd Desktop
gurukul@gurukul-ThinkCentre-M57e:~/Desktop$ cd CORBA
gurukul@gurukul-ThinkCentre-M57e:~/Desktop/CORBA$ idlj -fall
ReverseModule.idl\
>
gurukul@gurukul-ThinkCentre-M57e:~/Desktop/CORBA$ idlj -fall
ReverseModule.idl
gurukul@gurukul-ThinkCentre-M57e:~/Desktop/CORBA$ javac *.java
ReverseModule/*.java
Note: ReverseModule/ReversePOA.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
gurukul@gurukul-ThinkCentre-M57e:~/Desktop/CORBA$
gurukul@gurukul-ThinkCentre-M57e:~/Desktop/CORBA$ orbd -ORBInitialPort
1050&
[1] 3929
gurukul@gurukul-ThinkCentre-M57e:~/Desktop/CORBA$ java ReverseServer -
ORBInitialPort 1050& -ORBInitialHost localhost&
[2] 3948
[3] 3949
gurukul@gurukul-ThinkCentre-M57e:~/Desktop/CORBA$ -ORBInitialHost:
command not found
Reverse Object Created
Step1
Step2
Step3
Step4
Reverse Server reading and waiting....
```

Terminal 2_Clent:

```
gurukul@gurukul-ThinkCentre-M57e:~$ cd Desktop
gurukul@gurukul-ThinkCentre-M57e:~/Desktop$ cd CORBA
gurukul@gurukul-ThinkCentre-M57e:~/Desktop/CORBA$ java ReverseClient -
ORBInitialPort 1050 -ORBInitialHost localhost
Enter String=
Gunjan
Server Send najnuG
gurukul@gurukul-ThinkCentre-M57e:~/Desktop/CORBA$
```

**********Assignment 4(A)**********

Title: To develop any distributed algorithm for leader election using Bully Algorithm

```
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]) {
                System.out.println("0K");
            } 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("coordinate is " + i + "\nto
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: {
                    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);
    }
}
```

.....

OUTPUT

```
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
coordinate is4
to all
.........
1 up a process.
2.down a process
3 send a message
4.Exit
```

**********Assignment 4(B)**********

Title: To develop any distributed algorithm for leader election using Ring Algorithm

```
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++)
                  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 Enter the choice");
                  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;

                        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=

```
Enter the number of process :
5
Enter the id of process :
1
Enter the id of process :
2

Enter the id of process :
3
Enter the id of process :
4
Enter the id of process :
5
  [0] 1  [1] 2  [2] 3  [3] 4  [4] 5
 process 5select as co-ordinator

 Enter the choice

 1.Election 2.Quit
1

 Enter the Process number who initialsied election :
3

Process 4 send message to 1

Process 1 send message to 2

Process 2 send message to 3

Process 3 send message to 4

 process 4select as co-ordinator

 Enter the choice

 1.Election 2.Quit
```