ASSIGNMENT : 1

AddClient.java

```
import java.rmi.*;

public class AddClient {

public static void main(String args[]) {

try {

// Get reference to the remote object

String addServerURL = "rmi://" + args[0] + "/AddServer";

AddServerIntf addServerIntf =

(AddServerIntf) Naming.lookup(addServerURL);

System.out.println("The first number is: " + args[1]);

double d1 = Double.parseDouble(args[1]);

System.out.println("The second number is: " + args[2]);

double d2 = Double.parseDouble(args[2]);

// Invoke remote method to add numbers

System.out.println("The sum is: " + addServerIntf.add(d1, d2));

}

catch (Exception e) { System.out.println("Exception: "+ e);

}}}
```

AddServer.java

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

```
public class AddServer {
public static void main(String args[]) {
try {
//create remote object
AddServerImpl addServerImpl = new AddServerImpl();
//bind the remote object
Naming.rebind("AddServer",  addServerImpl);
}
catch (Exception e) {
System.out.println("Exception: "+ e);
}}}
```

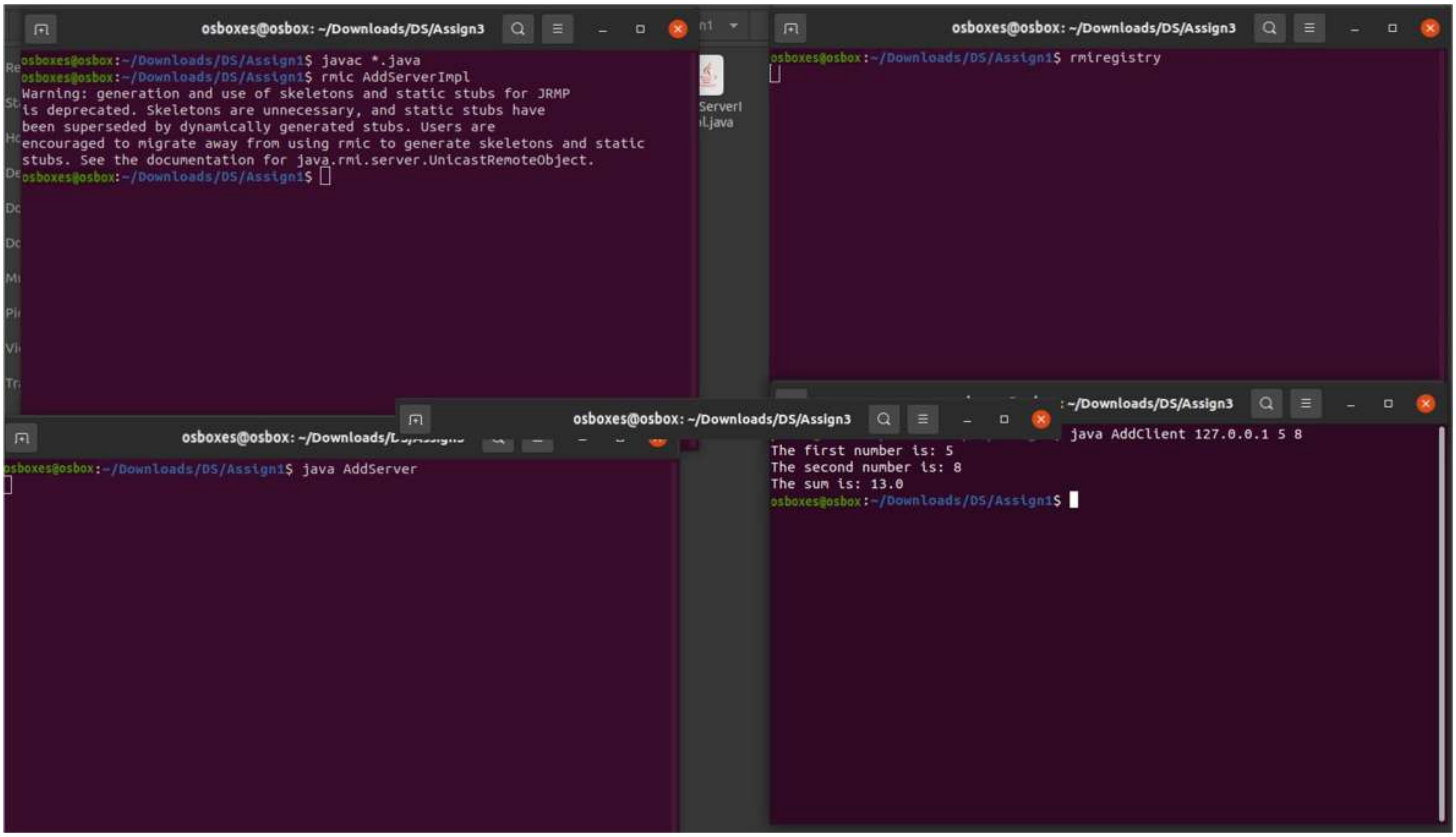
AddServerImpl.java

```
import java.rmi.*;
import java.rmi.server.*;
//class that implements the remote interface
public class AddServerImpl extends UnicastRemoteObject
implements AddServerIntf {
//constructor
public AddServerImpl() throws RemoteException {
}
//implement method declared in the interface
public double add(double d1, double d2) throws RemoteException {
 return d1 + d2; }}
```

AddServerIntf.java

```
import java.rmi.*;

public interface AddServerIntf extends Remote {

//method declaration

double add(double d1, double d2) throws RemoteException;

}
```

Output:



osboxes@osbox: ~/Downloads/DS/Assign3
osboxes@osbox: ~/Downloads/DS/Assign3
osboxes@osbox: ~/Downloads/DS/Assign3
osboxes@osbox: ~/Downloads/DS/Assign3
:~/Downloads/DS/Assign3
ServerI
l.java

```
osboxes@osbox:~/Downloads/DS/Assign1$ javac *.java
osboxes@osbox:~/Downloads/DS/Assign1$ rmic AddServerImpl
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.
osboxes@osbox:~/Downloads/DS/Assign1$
```

```
osboxes@osbox:~/Downloads/DS/Assign1$ rmiregistry
```

```
osboxes@osbox:~/Downloads/DS/Assign1$ java AddServer
```

```
 java AddClient 127.0.0.1 5 8
The first number is: 5
The second number is: 8
The sum is: 13.0
osboxes@osbox:~/Downloads/DS/Assign1$
```

ASSIGNMENT : 2

```
ReverseClient.java// client
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
        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();
        }}}
```

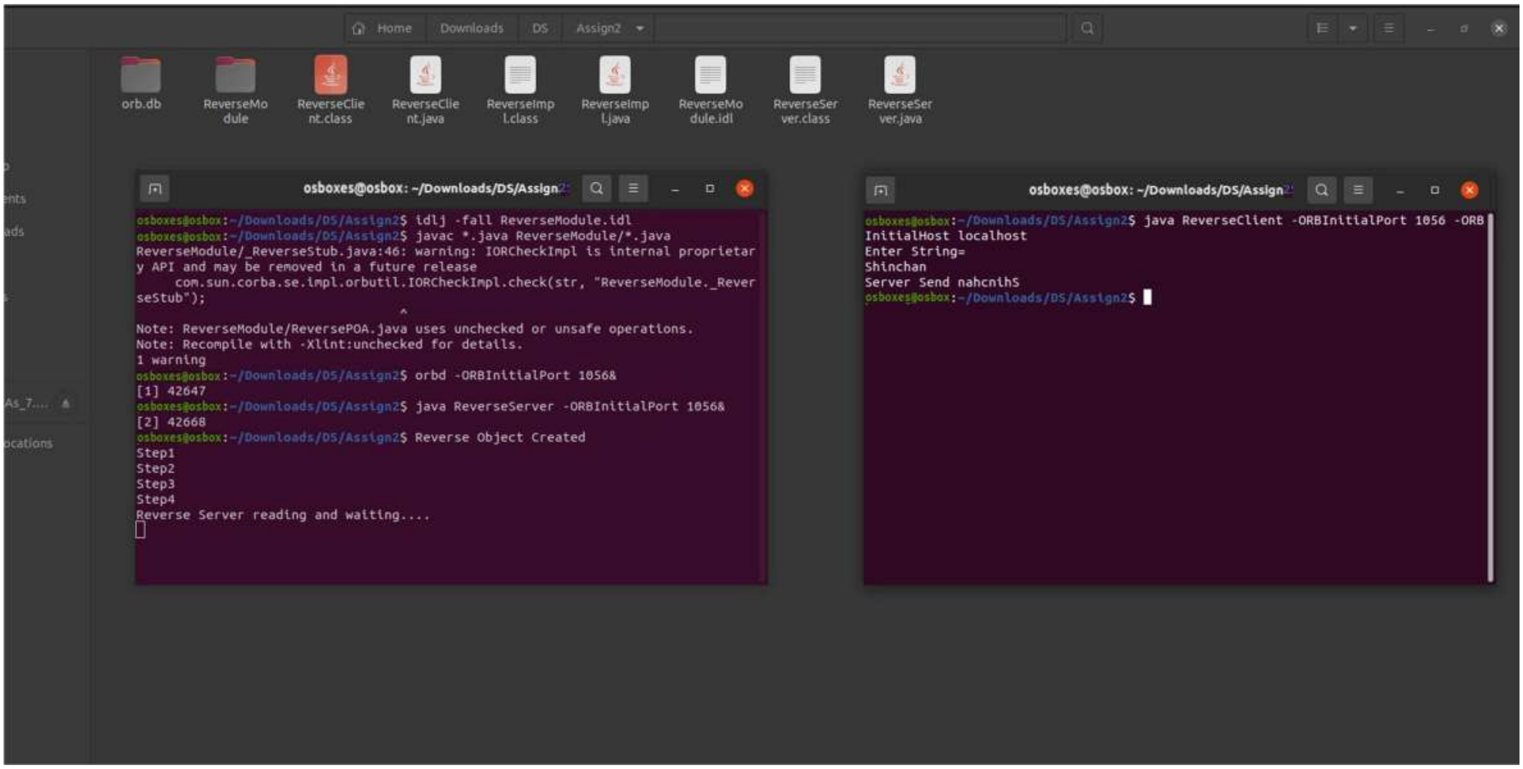
ReverseImpl.java

```
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));}}
```

Server.java

```
import ReverseModule.*;
import org.omg.CosNamig.*;
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 calculator 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:



```
osboxes@osbox:~/Downloads/DS/Assign2$ idlj -fall ReverseModule.idl
osboxes@osbox:~/Downloads/DS/Assign2$ javac *.java ReverseModule/*.java
ReverseModule/_ReverseStub.java:46: warning: IORCheckImpl is internal proprietar
y API and may be removed in a future release
     com.sun.corba.se.impl.orbutil.IORCheckImpl.check(str, "ReverseModule._Rever
seStub");
                                  ^
Note: ReverseModule/ReversePOA.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
1 warning
osboxes@osbox:~/Downloads/DS/Assign2$ orbd -ORBInitialPort 1056&
[1] 42647
osboxes@osbox:~/Downloads/DS/Assign2$ java ReverseServer -ORBInitialPort 1056&
[2] 42668
osboxes@osbox:~/Downloads/DS/Assign2$ Reverse Object Created
Step1
Step2
Step3
Step4
Reverse Server reading and waiting....
```

```
osboxes@osbox:~/Downloads/DS/Assign2$ java ReverseClient -ORBInitialPort 1056 -ORB
InitialHost localhost
Enter String=
Shinchan
Server Send nahcnihS
osboxes@osbox:~/Downloads/DS/Assign2$
```

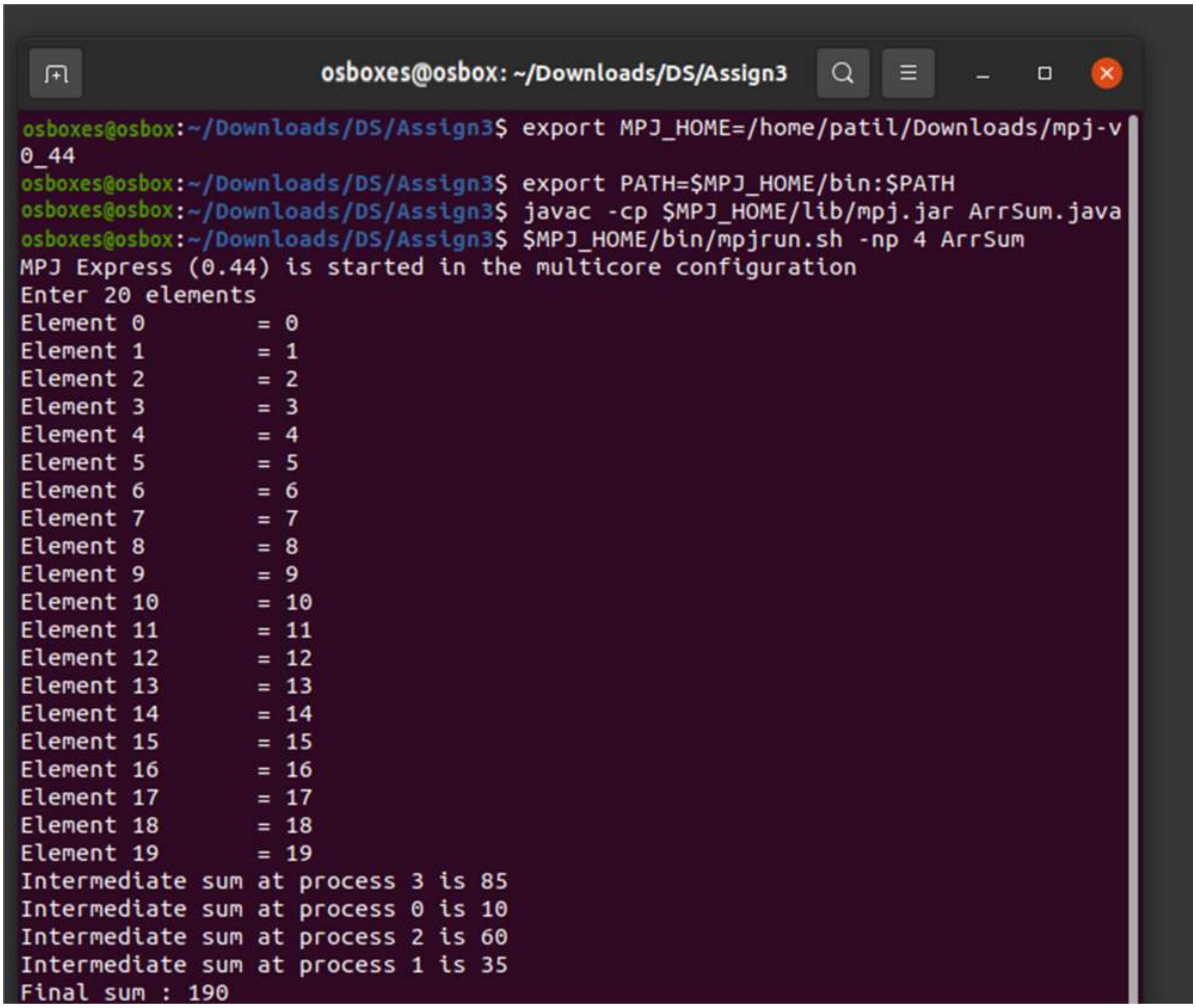
ASSIGNMENT : 3

ArrSum.java*// client*

```
import mpi.MPI;
import java.util.Scanner;
import mpi.*;
public class ArrSum {
    public static void main(String[] args) throws Exception{
        MPI.Init(args);
        int rank = MPI.COMM_WORLD.Rank();
        int size = MPI.COMM_WORLD.Size();
        int unitsize = 5;
        int root = 0;
        int send_buffer[] = null;
        //  1 process is expected to handle 4 elements
        send_buffer = new int [unitsize * size];
        int recieve_buffer[] = new int [unitsize];
        int new_recieve_buffer[] = new int [size];
        //  Set data for distribution
        if(rank == root) {
            int total_elements = unitsize * size;
            System.out.println("Enter " + total_elements + " elements");
    for(int i = 0; i < total_elements; i++) {
        System.out.println("Element " + i + "\t = " + i);
        send_buffer[i] = i;
    }
}
//  Scatter data to processes
MPI.COMM_WORLD.Scatter(
    send_buffer,
    0,
    unitsize,
    MPI.INT,
    recieve_buffer,
    0,
    unitsize,
    MPI.INT,
    root
);
//  Calculate sum at non root processes
//  Store result in first index of array
for(int i = 1; i < unitsize; i++) {
    recieve_buffer[0] += recieve_buffer[i];
}
System.out.println(
    "Intermediate sum at process " + rank + " is " + recieve_buffer[0]
);
```

```
    //  Gather data from processes

    MPI.COMM_WORLD.Gather(

      recieve_buffer,

      0,

      1,

      MPI.INT,

      new_recieve_buffer,

      0,

      1,

      MPI.INT,

      root

    );

    //  Aggregate output from all non root processes

    if(rank == root) {

     int total_sum = 0;

      for(int i = 0; i < size; i++) {

        total_sum += new_recieve_buffer[i];

      }

      System.out.println("Final sum : " + total_sum);

    }

    MPI.Finalize();

  }}
```

Output:



```
osboxes@osbox:~/Downloads/DS/Assign3$ export MPJ_HOME=/home/patil/Downloads/mpj-v
0_44
osboxes@osbox:~/Downloads/DS/Assign3$ export PATH=$MPJ_HOME/bin:$PATH
osboxes@osbox:~/Downloads/DS/Assign3$ javac -cp $MPJ_HOME/lib/mpj.jar ArrSum.java
osboxes@osbox:~/Downloads/DS/Assign3$ $MPJ_HOME/bin/mpjrun.sh -np 4 ArrSum
MPJ Express (0.44) is started in the multicore configuration
Enter 20 elements
Element 0        = 0
Element 1        = 1
Element 2        = 2
Element 3        = 3
Element 4        = 4
Element 5        = 5
Element 6        = 6
Element 7        = 7
Element 8        = 8
Element 9        = 9
Element 10       = 10
Element 11       = 11
Element 12       = 12
Element 13       = 13
Element 14       = 14
Element 15       = 15
Element 16       = 16
Element 17       = 17
Element 18       = 18
Element 19       = 19
Intermediate sum at process 3 is 85
Intermediate sum at process 0 is 10
Intermediate sum at process 2 is 60
Intermediate sum at process 1 is 35
Final sum : 190
```

ASSIGNMENT : 4

Client.py

```
# Python3 program imitating a client process

from timeit import default_timer as timer

from dateutil import parser

import threading

import  datetime

import socket

import time


# client thread function used to send time at client side

def startSendingTime(slave_client):


    while True:

        # provide server with clock time at the client

        slave_client.send(str(

                    datetime.datetime.now()).encode())


        print("Recent time sent successfully",

                                end = "\n\n")

        time.sleep(5)
```

```
# client thread function used to receive synchronized time
def startReceivingTime(slave_client):

    while True:
        # receive data from the server
        Synchronized_time = parser.parse(
                        slave_client.recv(1024).decode())

        print("Synchronized time at the client is: " + \
                                    str(Synchronized_time),
                                    end = "\n\n")

# function used to Synchronize client process time
def initiateSlaveClient(port = 8080):

    slave_client = socket.socket()

    # connect to the clock server on local computer
    slave_client.connect(('127.0.0.1', port))

    # start sending time to server
    print("Starting to receive time from server\n")
    send_time_thread = threading.Thread(
                    target = startSendingTime,
                args = (slave_client, ))
    send_time_thread.start()


    # start receiving synchronized from server
    print("Starting to receiving " + \
                    "synchronized time from server\n")
    receive_time_thread = threading.Thread(
                target = startReceivingTime,
                args = (slave_client, ))
    receive_time_thread.start()


# Driver function
if __name___== '__main__':


    # initialize the Slave / Client
    initiateSlaveClient(port = 8080)
```

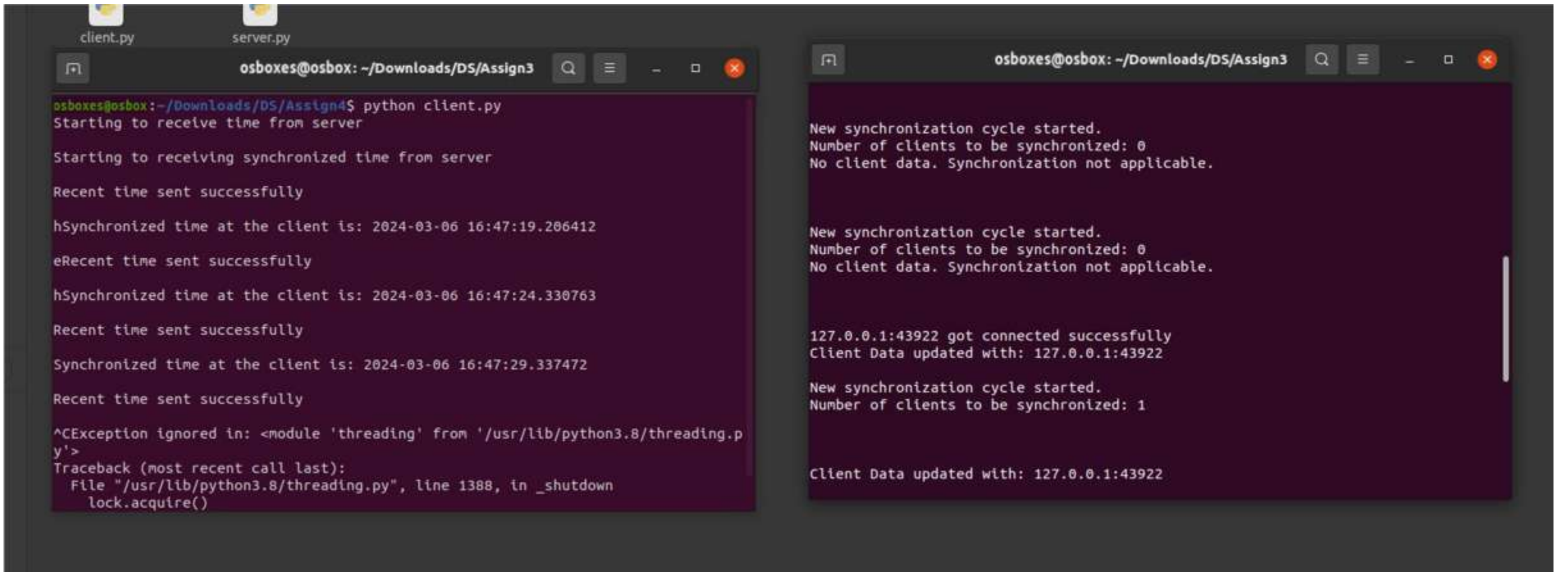
server.py

```
# Python3 program imitating a clock server
from dateutil import parser
import threading
import datetime
import socket
import time
```

```
# datastructure used to store client address and clock data
client_data = {}


''' nested thread function used to receive
    clock time from a connected client '''
def startReceivingClockTime(connector, address):


    while True:
        # receive clock time
        clock_time_string = connector.recv(1024).decode()
        clock_time = parser.parse(clock_time_string)
        clock_time_diff = datetime.datetime.now() - \
                                               clock_time


        client_data[address] = {
                    "clock_time"     : clock_time,
                    "time_difference" : clock_time_diff,
                    "connector"  : connector
                    }


        print("Client Data updated with: "+ str(address),
                                           end = "\n\n")
        time.sleep(5)


''' master thread function used to open portal for
```

```
    accepting clients over given port '''
def startConnecting(master_server):


    # fetch clock time at slaves / clients
    while True:
        # accepting a client / slave clock client
        master_slave_connector, addr = master_server.accept()
        slave_address = str(addr[0]) + ":" + str(addr[1])


        print(slave_address + " got connected successfully")


        current_thread = threading.Thread(
                        target = startReceivingClockTime,
                        args = (master_slave_connector,
                                       slave_address, ))
        current_thread.start()


# subroutine function used to fetch average clock difference
def getAverageClockDiff():


    time_difference_list = list(client['time_difference']
                                for client_addr, client
                                    in client_data.items())
    sum_of_clock_difference = sum(time_difference_list, \
                               datetime.timedelta(0, 0))

    average_clock_difference = sum_of_clock_difference \
                                       / len(client_data)

    return average_clock_difference

''' master sync thread function used to generate
    cycles of clock synchronization in the network '''
def synchronizeAllClocks():

    while True:

        print("New synchronization cycle started.")
        print("Number of clients to be synchronized: " + \
                                   str(len(client_data)))

        if len(client_data) > 0:

            average_clock_difference = getAverageClockDiff()

            for client_addr, client in client_data.items():
                try:
                    synchronized_time = \
                        datetime.datetime.now() + \
                                    average_clock_difference

                    client['connector'].send(str(
                            synchronized_time).encode())

                except Exception as e:
                    print("Something went wrong while " + \
                        "sending synchronized time " + \
                        "through " + str(client_addr))

        else :
            print("No client data." + \
                        " Synchronization not applicable.")

        print("\n\n")

        time.sleep(5)

# function used to initiate the Clock Server / Master Node
def initiateClockServer(port = 8080):

    master_server = socket.socket()
    master_server.setsockopt(socket.SOL_SOCKET,
                                socket.SO_REUSEADDR, 1)
    print("Socket at master node created successfully\n")
    master_server.bind(('', port))
    # Start listening to requests
    master_server.listen(10)
    print("Clock server started...\n")
    # start making connections
    print("Starting to make connections...\n")
    master_thread = threading.Thread(
                        target = startConnecting,
                        args = (master_server, ))
    master_thread.start()
    # start synchronization
    print("Starting synchronization parallelly...\n")
    sync_thread = threading.Thread(
                        target = synchronizeAllClocks,
                        args = ())
    sync_thread.start()
# Driver function
if __name___== '__main__':

    # Trigger the Clock Server
    initiateClockServer(port = 8080)
```

Output:



```
osboxes@osbox:~/Downloads/DS/Assign4$ python client.py
Starting to receive time from server

Starting to receiving synchronized time from server

Recent time sent successfully

hSynchronized time at the client is: 2024-03-06 16:47:19.206412

eRecent time sent successfully

hSynchronized time at the client is: 2024-03-06 16:47:24.330763

Recent time sent successfully

Synchronized time at the client is: 2024-03-06 16:47:29.337472

Recent time sent successfully

^CException ignored in: <module 'threading' from '/usr/lib/python3.8/threading.p
y'>
Traceback (most recent call last):
  File "/usr/lib/python3.8/threading.py", line 1388, in _shutdown
    lock.acquire()
```

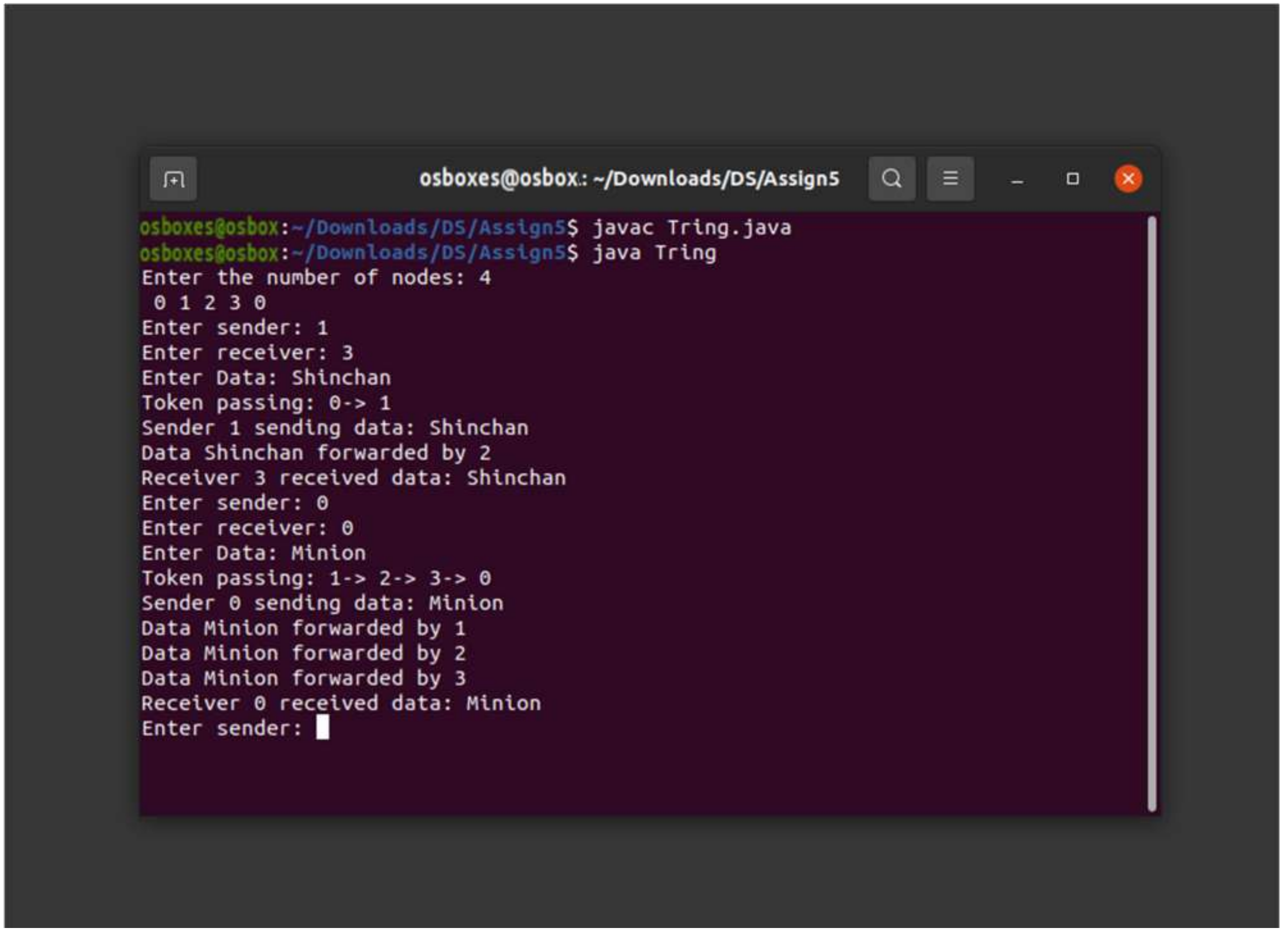
```
New synchronization cycle started.
Number of clients to be synchronized: 0
No client data. Synchronization not applicable.



New synchronization cycle started.
Number of clients to be synchronized: 0
No client data. Synchronization not applicable.



127.0.0.1:43922 got connected successfully
Client Data updated with: 127.0.0.1:43922

New synchronization cycle started.
Number of clients to be synchronized: 1



Client Data updated with: 127.0.0.1:43922
```

ASSIGNMENT : 5

Tring.java

```
import java.util.Scanner;


class Tring {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);


        System.out.print("Enter the number of nodes: ");
        int n = sc.nextInt();


        // Decides the number of nodes forming the ring
        int token = 0;


        for (int i = 0; i < n; i++)
            System.out.print(" " + i);
        System.out.println(" " + 0);


        try {
            while (true) {
                System.out.print("Enter sender: ");
            int s = sc.nextInt();
            System.out.print("Enter receiver: ");
            int r = sc.nextInt();
            System.out.print("Enter Data: ");
            String d = sc.next();

            System.out.print("Token  passing:");
             //current token not equal to sender, increment i by 1 and j by j+1%n

            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: " + d);

            // start forwarding from node after sender until it becomes equal to receiver and increment  by i+1%n
            for (int i = (s + 1) % n; i != r; i = (i + 1) % n) {
            System.out.println("Data " + d + " forwarded by " + i);
            }
            System.out.println("Receiver " + r + " received data: " + d);
            token = s;
          }
        } catch (Exception e) {
          System.out.println("Error occurred: " + e.getMessage());
        }
    }
}
```

Output:



```
osboxes@osbox:~/Downloads/DS/Assign5$ javac Tring.java
osboxes@osbox:~/Downloads/DS/Assign5$ java Tring
Enter the number of nodes: 4
 0 1 2 3 0
Enter sender: 1
Enter receiver: 3
Enter Data: Shinchan
Token passing: 0-> 1
Sender 1 sending data: Shinchan
Data Shinchan forwarded by 2
Receiver 3 received data: Shinchan
Enter sender: 0
Enter receiver: 0
Enter Data: Minion
Token passing: 1-> 2-> 3-> 0
Sender 0 sending data: Minion
Data Minion forwarded by 1
Data Minion forwarded by 2
Data Minion forwarded by 3
Receiver 0 received data: Minion
Enter sender: 
```

ASSIGNMENT : 6

Bully.java

```
import java.util.*;


public class Bully {
    int coordinator;
    int max_processes;
    boolean processes[];


    public Bully(int max) {
        max_processes = max;
        processes = new boolean[max_processes];
        coordinator = max;


        System.out.println("Creating processes..");
        for(int i = 0; i < max; i++) {
            processes[i] = true;
            System.out.println("P"+ (i+1) + " created");
        }
        System.out.println("Process P" + coordinator + " is the coordinator");


    }
```

```
void displayProcesses() {
    for(int i = 0; i < max_processes; i++) {
        if(processes[i]) {
            System.out.println("P" + (i+1) + " is up");
        } else {
            System.out.println("P" + (i+1) + " is down");
        }
    }
    System.out.println("Process P" + coordinator + " is the coordinator");
}

void upProcess(int process_id) {
    if(!processes[process_id - 1]) {
        processes[process_id - 1] = true;
        System.out.println("Process " + process_id + " is now up.");
    } else {
        System.out.println("Process " + process_id + " is already up.");
    }
}

void downProcess(int process_id) {
    if(!processes[process_id - 1]) {
        System.out.println("Process " + process_id + " is already down.");
    } else {
        processes[process_id - 1] = false;
        System.out.println("Process " + process_id + " is down.");
    }
}

void runElection(int process_id) {
    coordinator = process_id;
    boolean keepGoing = true;

    for(int i = process_id; i < max_processes && keepGoing; i++) {
        System.out.println("Election message sent from process " + process_id + " to process " + (i+1));

        if(processes[i]) {
            keepGoing = false;
            runElection(i + 1);
        }
    }
}

public static void main(String args[]) {
    Bully bully = null;
    int max_processes = 0, process_id = 0;
    int choice = 0;
    Scanner sc = new Scanner(System.in);

    while(true) {
```

```
System.out.println("Bully Algorithm");
System.out.println("1. Create processes");
System.out.println("2. Display processes");
System.out.println("3. Up a process");
System.out.println("4. Down a process");
System.out.println("5. Run election algorithm");
System.out.println("6. Exit Program");
System.out.print("Enter your choice:- ");
choice = sc.nextInt();

switch(choice) {
    case 1:
        System.out.print("Enter the number of processes:- ");
        max_processes = sc.nextInt();
        bully = new Bully(max_processes);
        break;
    case 2:
        bully.displayProcesses();
        break;
    case 3:
        System.out.print("Enter the process number to up:- ");
        process_id = sc.nextInt();
        bully.upProcess(process_id);
        break;
    case 4:
          System.out.print("Enter the process number to down:- ");
          process_id = sc.nextInt();
          bully.downProcess(process_id);
          break;
        case 5:
          System.out.print("Enter the process number which will perform election:- ");
          process_id = sc.nextInt();
          bully.runElection(process_id);
          bully.displayProcesses();
          break;
        case 6:
          System.exit(0);
          break;
        default:
          System.out.println("Error in choice. Please try again.");
          break;
      }}}}
```

Ring.java

```
import java.util.*;


public class Ring {
  int max_processes;
  int coordinator;
  boolean processes[];
  ArrayList<Integer> pid;
```

```
public Ring(int max) {
    coordinator = max;
    max_processes = max;
    pid = new ArrayList<Integer>();
    processes = new boolean[max];

    for(int i = 0; i < max; i++) {
        processes[i] = true;
        System.out.println("P" + (i+1) + " created.");
    }
    System.out.println("P" + (coordinator) + " is the coordinator");
}

void displayProcesses() {
    for(int i = 0; i < max_processes; i++) {
        if(processes[i])
            System.out.println("P" + (i+1) + " is up.");
        else
            System.out.println("P" + (i+1) + " is down.");
    }
    System.out.println("P" + (coordinator) + " is the coordinator");
}

void upProcess(int process_id) {
    if(!processes[process_id-1]) {
        processes[process_id-1] = true;
        System.out.println("Process P" + (process_id) + " is up.");
    } else {
        System.out.println("Process P" + (process_id) + " is already up.");
    }
}

void downProcess(int process_id) {
    if(!processes[process_id-1]) {
        System.out.println("Process P" + (process_id) + " is already down.");
    } else {
        processes[process_id-1] = false;
        System.out.println("Process P" + (process_id) + " is down.");
    }
}

void displayArrayList(ArrayList<Integer> pid) {
    System.out.print("[ ");
    for(Integer x : pid) {
        System.out.print(x + " ");
    }
    System.out.print(" ]\n");
}
```

```
    void initElection(int process_id) {
        if(processes[process_id-1]) {
            pid.add(process_id);

            int temp = process_id;

            System.out.print("Process P" + process_id + " sending the following list:- ");
            displayArrayList(pid);

            while(temp != process_id - 1) {
                if(processes[temp]) {
                    pid.add(temp+1);
                    System.out.print("Process P" + (temp + 1) + " sending the following list:- ");
                    displayArrayList(pid);
                }
                temp = (temp + 1) % max_processes;
            }
            coordinator = Collections.max(pid);
            System.out.println("Process P" + process_id + " has declared P" + coordinator + " as the coordinator");
            pid.clear();
        }
    }

    public static void main(String args[]) {
        Ring ring = null;
```

```
int max_processes = 0, process_id = 0;

int choice = 0;

Scanner sc = new Scanner(System.in);


while(true) {

   System.out.println("Ring Algorithm");

   System.out.println("1. Create processes");

   System.out.println("2. Display processes");

   System.out.println("3. Up a process");

   System.out.println("4. Down a process");

   System.out.println("5. Run election algorithm");

   System.out.println("6. Exit Program");

   System.out.print("Enter your choice:- ");

   choice = sc.nextInt();


   switch(choice) {

      case 1:

         System.out.print("Enter the total number of processes:- ");

         max_processes = sc.nextInt();

         ring = new Ring(max_processes);

         break;

      case 2:

         ring.displayProcesses();

         break;

      case 3:
```

```
      System.out.print("Enter the process to up:- ");

      process_id = sc.nextInt();

      ring.upProcess(process_id);

      break;

    case 4:

      System.out.print("Enter the process to down:- ");

      process_id = sc.nextInt();

      ring.downProcess(process_id);

      break;

    case 5:

      System.out.print("Enter the process which will initiate election:- ");

      process_id = sc.nextInt();

      ring.initElection(process_id);

      break;

    case 6:

      System.exit(0);

      break;

    default:

      System.out.println("Error in choice. Please try again.");

      break;}} }}
```

Output:

BULLY

```
osboxes@osbox:~/Downloads/DS/Assign6$ javac  Bully.java
osboxes@osbox:~/Downloads/DS/Assign6$ java Bully Bully Algorithm
```

```
1. Create processes
2. Display processes
3. Up a process
4. Down a process
5. Run election algorithm
6. Exit Program
Enter your choice:- 1
Enter the number of processes:- 4
Creating processes..
P1 created
P2 created
P3 created
P4 created
Process P4 is the coordinator
Bully Algorithm
1. Create processes
2. Display processes
3. Up a process
4. Down a process
5. Run election algorithm
6. Exit Program
Enter your choice:- 2
P1 is up
P2 is up
P3 is up
```

```
P4 is up
Process P4 is the coordinator
Bully Algorithm
1. Create processes
2. Display processes
3. Up a process
4. Down a process
5. Run election algorithm
6. Exit Program
Enter your choice:- 4
Enter the process number to down:- 2
Process 2 is down.
Bully Algorithm
1. Create processes
2. Display processes
3. Up a process
4. Down a process
5. Run election algorithm
6. Exit Program
Enter your choice:- 2
P1 is up
P2 is down
P3 is up
P4 is up
Process P4 is the coordinator
```

```
Bully Algorithm
1. Create processes
2. Display processes
3. Up a process
4. Down a process
5. Run election algorithm
6. Exit Program
Enter your choice:- 5
Enter the process number which will perform election:- 3
Election message sent from process 3 to process 4
P1 is up
P2 is down
P3 is up
P4 is up
Process P4 is the coordinator
Bully Algorithm
1. Create processes
2. Display processes
3. Up a process
4. Down a process
5. Run election algorithm
6. Exit Program
Enter your choice:- 6
```

RING

```
osboxes@osbox:~/Downloads/DS/Assign6$ javac Ring.java

osboxes@osbox:~/Downloads/DS/Assign6$ java Ring

Ring Algorithm

1. Create processes

2. Display processes

3. Up a process

4. Down a process

5. Run election algorithm

6. Exit Program

Enter your choice:- 1

Enter the total number of processes:- 4

P1 created.

P2 created.

P3 created.

P4 created.

P4 is the coordinator

Ring Algorithm

1. Create processes

2. Display processes

3. Up a process

4. Down a process

5. Run election algorithm

6. Exit Program

Enter your choice:- 2

P1 is up.
```

```
P2 is up.

P3 is up.

P4 is up.

P4 is the coordinator

Ring Algorithm

1. Create processes

2. Display processes

3. Up a process

4. Down a process

5. Run election algorithm

6. Exit Program

Enter your choice:- 4

Enter the process to down:- 2

Process P2 is down.

Ring Algorithm

1. Create processes

2. Display processes

3. Up a process

4. Down a process

5. Run election algorithm

6. Exit Program

Enter your choice:- 2

P1 is up.

P2 is down.

P3 is up.
```

```
P4 is up.
P4 is the coordinator
Ring Algorithm
1. Create processes
2. Display processes
3. Up a process
4. Down a process
5. Run election algorithm
6. Exit Program
Enter your choice:- 5
Enter the process which will initiate election:- 3
Process P3 sending the following list:- [ 3  ]
Process P4 sending the following list:- [ 3 4  ]
Process P1 sending the following list:- [ 3 4 1  ]
Process P3 has declared P4 as the coordinator
Ring Algorithm
1. Create processes
2. Display processes
3. Up a process
4. Down a process
5. Run election algorithm
6. Exit Program
Enter your choice:- 6
```

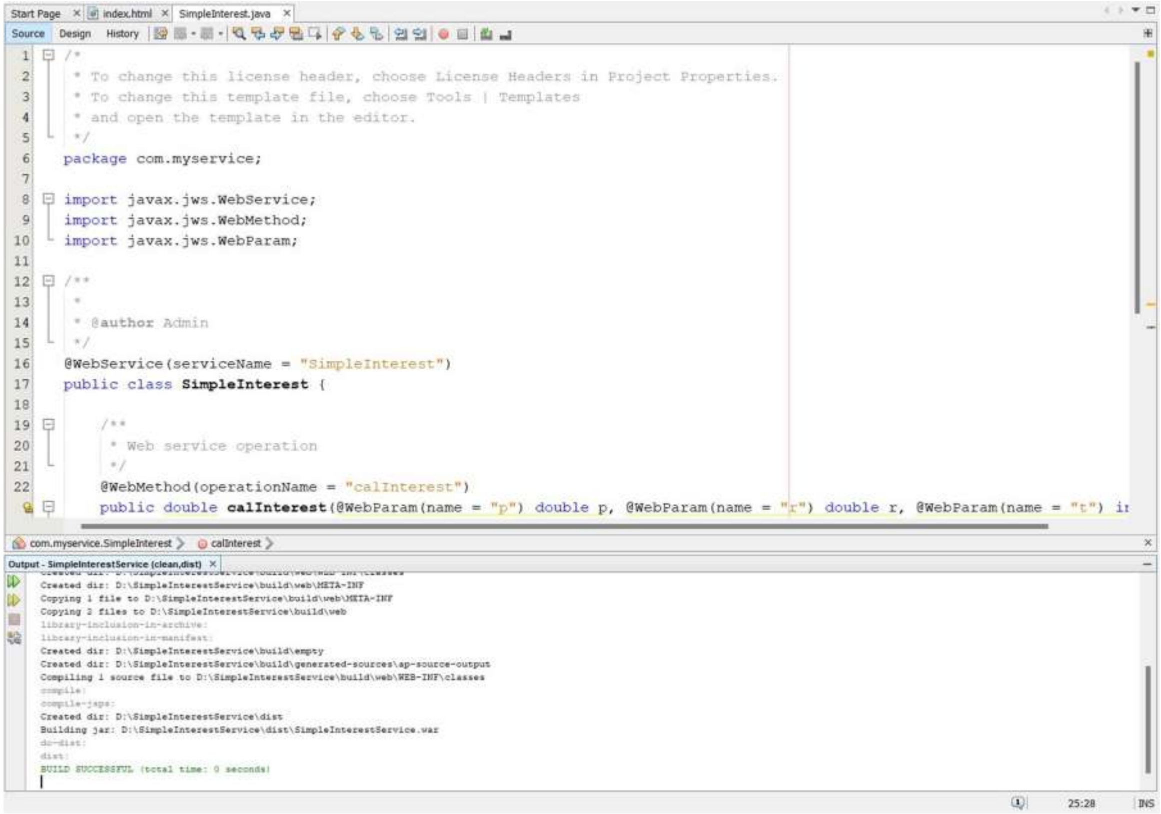
ASSIGNMENT-7

Code:

SimpleInterest.java

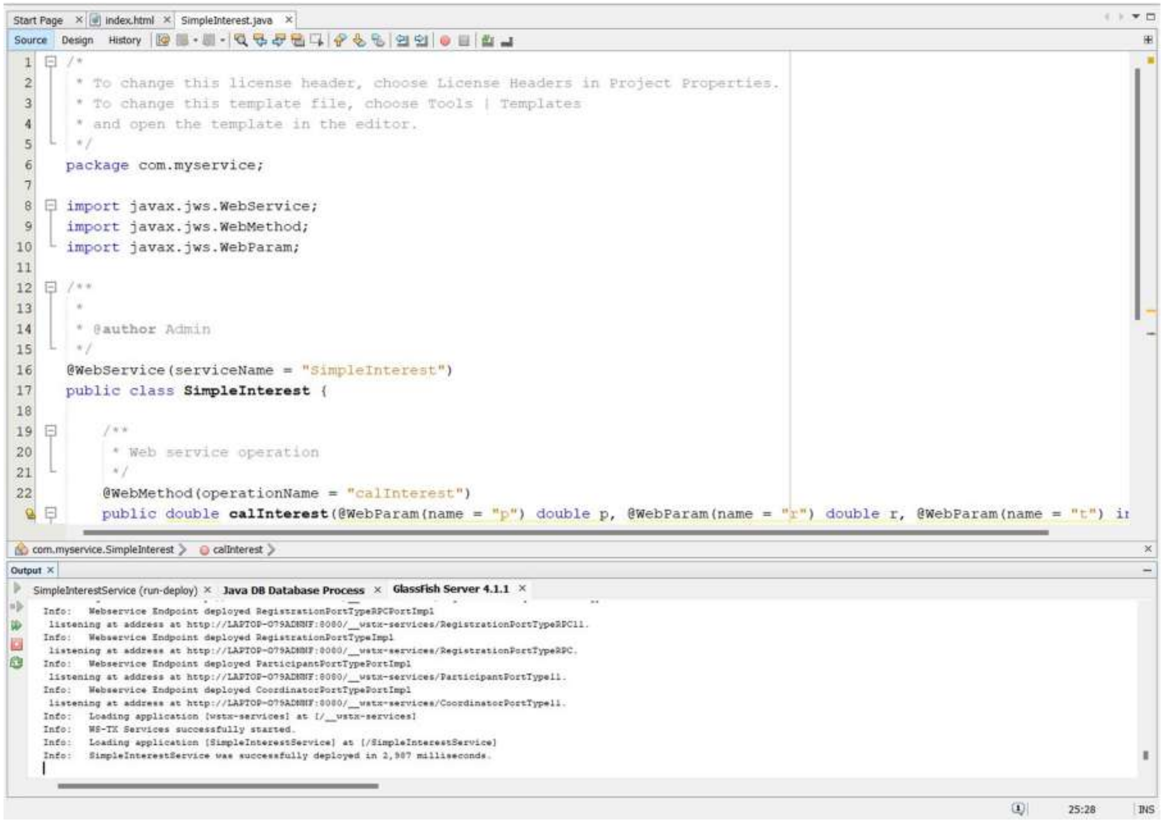
```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.myservice;

import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

/**
 *
 * @author Admin
 */
@WebService(serviceName = "SimpleInterest")
public class SimpleInterest {

    /**
     * Web service operation
     */
    @WebMethod(operationName = "callInterest")
    public double callInterest(@WebParam(name = "p") double p, @WebParam(name = "r")
double r, @WebParam(name = "t") int t) {
        //TODO write your implementation code here:
        return (p*r*t)/100;
    }

    /**
     * This is a sample web service operation
     */
}
```

Screenshots:



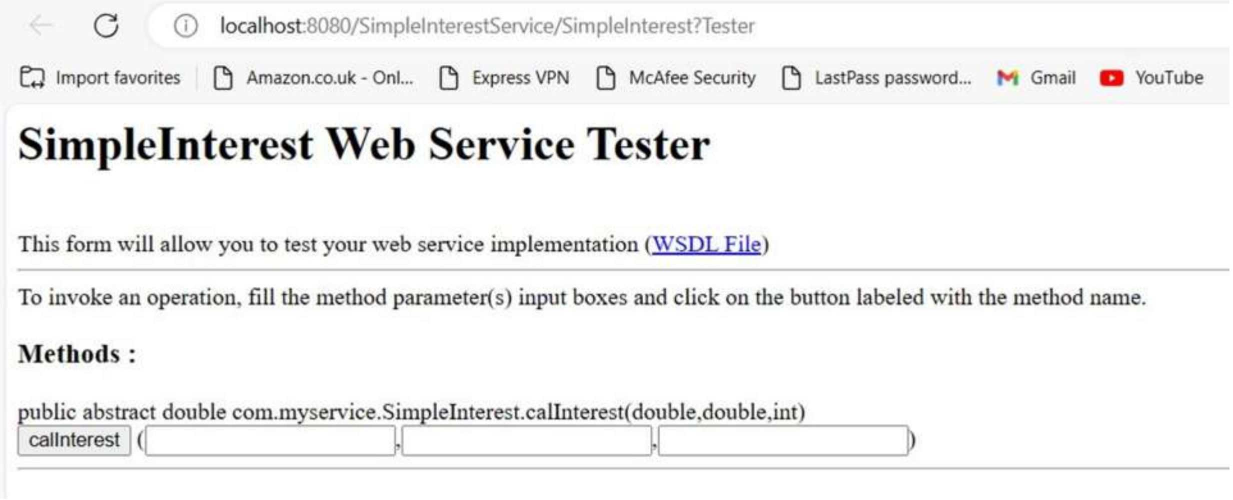
```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.myservice;

import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

/**
 *
 * @author Admin
 */
@WebService(serviceName = "SimpleInterest")
public class SimpleInterest {

    /**
     * Web service operation
     */
    @WebMethod(operationName = "calInterest")
    public double calInterest(@WebParam(name = "p") double p, @WebParam(name = "r") double r, @WebParam(name = "t") i
```

```
Created dir: D:\SimpleInterestService\build\web\META-INF
Copying 1 file to D:\SimpleInterestService\build\web\META-INF
Copying 2 files to D:\SimpleInterestService\build\web
library-inclusion-in-archive:
library-inclusion-in-manifest:
Created dir: D:\SimpleInterestService\build\empty
Created dir: D:\SimpleInterestService\build\generated-sources\ap-source-output
Compiling 1 source file to D:\SimpleInterestService\build\web\WEB-INF\classes
compile:
compile-jsps:
Created dir: D:\SimpleInterestService\dist
Building jar: D:\SimpleInterestService\dist\SimpleInterestService.war
do-dist:
dist:
BUILD SUCCESSFUL (total time: 0 seconds)
```



```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 com.myservice;

import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

/**
 *
 * @author Admin
 */
@WebService(serviceName = "SimpleInterest")
public class SimpleInterest {

    /**
     * Web service operation
     */
    @WebMethod(operationName = "calInterest")
    public double calInterest(@WebParam(name = "p") double p, @WebParam(name = "r") double r, @WebParam(name = "t") i
```

```
Info:   Webservice Endpoint deployed RegistrationPortTypeRPCPortImpl
 listening at address at http://LAPTOP-079ADNNF:8080/__wstx-services/RegistrationPortTypeRPC11.
Info:   Webservice Endpoint deployed RegistrationPortTypeImpl
 listening at address at http://LAPTOP-079ADNNF:8080/__wstx-services/RegistrationPortTypeRPC.
Info:   Webservice Endpoint deployed ParticipantPortTypePortImpl
 listening at address at http://LAPTOP-079ADNNF:8080/__wstx-services/ParticipantPortType11.
Info:   Webservice Endpoint deployed CoordinatorPortTypePortImpl
 listening at address at http://LAPTOP-079ADNNF:8080/__wstx-services/CoordinatorPortType11.
Info:   Loading application [wstx-services] at [/__wstx-services]
Info:   WS-TX Services successfully started.
Info:   Loading application [SimpleInterestService] at [/SimpleInterestService]
Info:   SimpleInterestService was successfully deployed in 2,987 milliseconds.
```



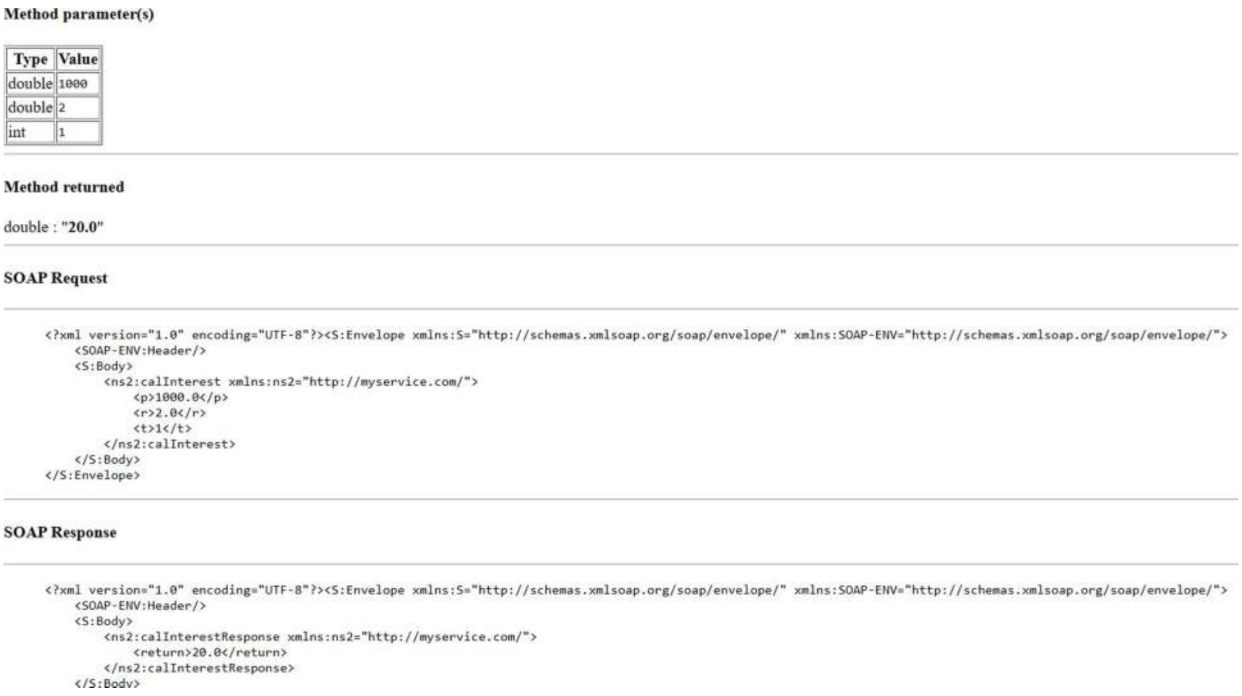
SimpleInterest Web Service Tester

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 double com.myservice.SimpleInterest.calInterest(double,double,int)



Method parameter(s)

Type	Value
double	1000
double	2
int	1

Method returned

double : **"20.0"**

SOAP Request

```
<?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:calInterest xmlns:ns2="http://myservice.com/">
            <p>1000.0</p>
            <r>2.0</r>
            <t>1</t>
        </ns2:calInterest>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?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:calInterestResponse xmlns:ns2="http://myservice.com/">
            <return>20.0</return>
        </ns2:calInterestResponse>
    </S:Body>
```