**Assignment no:2**

**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

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";

//Helper class provides narrow method that cast corba object reference (ref) into the java interface

// System.out.println("Step2");

// Look ups "Reverse" in the naming context

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 portable object adaptor (BOA/POA) connects client request using object reference

//uses orb method as resolve\_initial\_references

POA rootPOA = POAHelper.narrow(orb.resolve\_initial\_references("RootPOA"));

rootPOA.the\_POAManager().activate();

// creating an object of ReverseImpl class

ReverseImpl rvr = new ReverseImpl();

//server consist of 2 classes ,servent and server. The servent is the subclass of ReversePOA which is generated by the idlj compiler

// The servent ReverseImpl is the implementation of the ReverseModule idl interface

// get the object reference from the servant class

//use root POA class and its method servant\_to\_reference

org.omg.CORBA.Object ref = rootPOA.servant\_to\_reference(rvr);

// System.out.println("Step1");

Reverse h\_ref = ReverseModule.ReverseHelper.narrow(ref);// Helper class provides narrow method that cast corba object reference (ref) into the java interface

// System.out.println("Step2");

// orb layer uses resolve\_initial\_references method to take initial reference as NameService

org.omg.CORBA.Object objRef = orb.resolve\_initial\_references("NameService");

//Register new object in the naming context under the Reverse

// 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);

//Server run and waits for invocations of the new object from the client

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));

}

}

**Output:**





