

CODE :

1)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";

            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();
    }
}
}

```

2)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);

            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);

    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");

    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();
}
}
}

```

3)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));

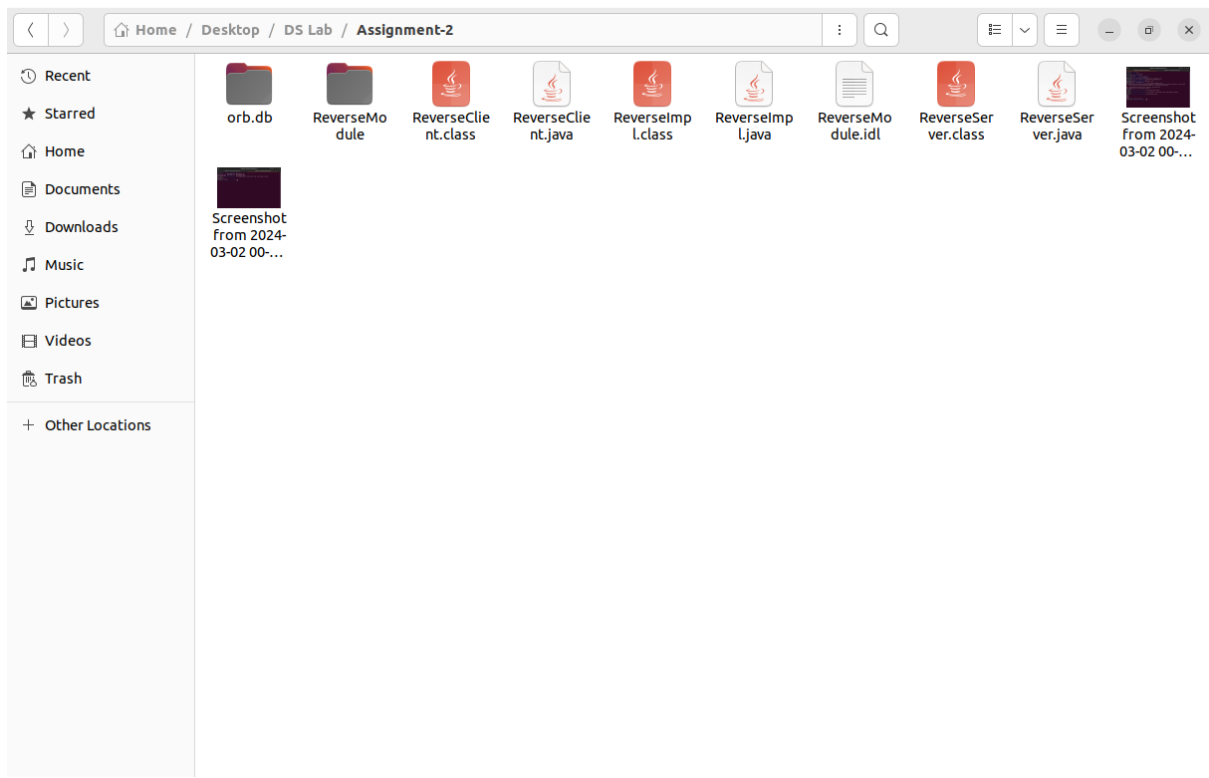
    }

}
```

4)ReverseModule.idl :

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

OUTPUT:



```
aatif@aatif: ~/Desktop/DS Lab/Assignment-2

aatif@aatif: ~/Desktop/DS Lab/Assignment-2
aatif@aatif: ~/Desktop$ ls
'DS Lab'
aatif@aatif:~/Desktop$ cd DS\ Lab/
aatif@aatif:~/Desktop/DS Lab$ ls
'Asg 3 'Asg 3.docx'' Assignment-1 Assignment-2
aatif@aatif:~/Desktop/DS Lab$ cd Assignment-2
aatif@aatif:~/Desktop/DS Lab/Assignment-2$ ls
ReverseClient.java ReverseImpl.java ReverseModule.idl ReverseServer.java
aatif@aatif:~/Desktop/DS Lab/Assignment-2$ idlj -fall ReverseModule.idl
aatif@aatif:~/Desktop/DS Lab/Assignment-2$ ls
ReverseClient.java ReverseModule ReverseServer.java
ReverseImpl.java ReverseModule.idl
aatif@aatif:~/Desktop/DS Lab/Assignment-2$ javac *.java ReverseModule/*.java
ReverseModule/_ReverseStub.java:46: warning: IORCheckImpl is internal proprietary API and may be removed in a future release
    com.sun.corba.se.impl.orbutil.IORCheckImpl.check(str, "ReverseModule._ReverseStub");
                                   ^
Note: ReverseModule/ReversePOA.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
1 warning
aatif@aatif:~/Desktop/DS Lab/Assignment-2$ orbd -ORBInitialPort 1050&
[1] 2651
aatif@aatif:~/Desktop/DS Lab/Assignment-2$ java ReverseServer -ORBInitialPort 1050 -ORBInitialHost localhost&
[2] 2676
aatif@aatif:~/Desktop/DS Lab/Assignment-2$ Reverse Object Created
Step1
Step2
Step3
Step4
Reverse Server reading and waiting...
aatif@aatif:~/Desktop/DS Lab/Assignment-2$
```

```
aatif@aatif: ~/Desktop/DS Lab/Assignment-2
aatif@aatif: ~/Desktop/DS Lab/Assignment-2
aatif@aatif:~/Desktop/DS Lab/Assignment-2$ ls
orb.db          ReverseImpl.class  ReverseModule.idl
ReverseClient.class  ReverseImpl.java  ReverseServer.class
ReverseClient.java   ReverseModule      ReverseServer.java
aatif@aatif:~/Desktop/DS Lab/Assignment-2$ java ReverseClient -ORBInitialPort 1050 -ORBInitialHost localhost
Enter String=
hey Aatif
Server Send  fitaA yeh
aatif@aatif:~/Desktop/DS Lab/Assignment-2$
```