

ReverseClient.java

```
1 // client
2
3 import ReverseModule.*;
4
5 import org.omg.CosNaming.*;
6
7 import org.omg.CosNaming.NamingContextPackage.*;
8
9 import org.omg.CORBA.*;
10
11 import java.io.*;
12 class ReverseClient
13 {
14
15     public static void main(String args[])
16     {
17
18         Reverse ReverseImpl=null;
19
20         try
21         {
22             // initialize the ORB
23
24             org.omg.CORBA.ORB orb = org.omg.CORBA.ORB.init(args, null);
25
26             org.omg.CORBA.Object objRef = orb.resolve_initial_references ("NameService");
27
28             NamingContextExt ncRef = NamingContextExtHelper.narrow (objRef);
29
30             String name = "Reverse";
31
32             ReverseImpl = ReverseHelper.narrow(ncRef.resolve_str(name));
33
34             System.out.println("Enter String=");
35
36             BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
37
38             String str = br.readLine();
39
40             String tempStr = ReverseImpl.reverse_string(str);
41
42             System.out.println(tempStr);
43
44         }
45     }
46 }
47
48
```

```
49  
50  
51  
52     catch (Exception e)  
53  
54     {  
55  
56         e.printStackTrace();  
57  
58     }  
59  
60 }  
61  
62 }
```

ReverseImpl.java

```
1  import ReverseModule.ReversePOA;
2
3  import java.lang.String;
4
5  class ReverseImpl extends ReversePOA
6
7  {
8
9      ReverseImpl()
10
11      {
12
13          super();
14
15          System.out.println("Reverse Object Created");
16
17      }
18
19      public String reverse_string(String name)
20
21      {
22
23          StringBuffer str=new StringBuffer(name);
24
25          str.reverse();
26
27          return (("Server Send "+str));
28
29      }
30
31 }
```

ReverseModule.idl

```
1 module ReverseModule
2 {
3   interface Reverse
4   {
5     string reverse_string(in string str);
6   };
7 };
```

ReverseServer.java

```
1  import ReverseModule.*;
2
3  import org.omg.CosNaming.*;
4
5  import org.omg.CosNaming.NamingContextPackage.*;
6
7  import org.omg.CORBA.*;
8
9  import org.omg.PortableServer.*;
10
11
12
13  class ReverseServer
14
15  {
16
17      public static void main(String[] args)
18
19      {
20
21          try
22
23          { // initialize the ORB
24
25              org.omg.CORBA. ORB orb = org.omg.CORBA.ORB.init(args, null);
26
27
28
29              // initialize the BOA/POA
30
31              POA rootPOA = POAHelper.narrow(orb.resolve_initial_references("RootPOA"));
32
33              rootPOA.the_POAManager().activate();
34
35              // creating the calculator object
36
37              ReverseImpl rvr = new ReverseImpl();
38
39
40
41              // get the object reference from the servant class
42
43              org.omg.CORBA.Object ref = rootPOA.servant_to_reference(rvr);
44
45
46
47              System.out.println("Step1");
48
```

```
49     Reverse h_ref = ReverseModule.ReverseHelper.narrow(ref);
50
51     System.out.println("Step2");
52
53
54
55     org.omg.CORBA.Object objRef = orb.resolve_initial_references("NameService");
56
57
58
59     System.out.println("Step3");
60
61     NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef);
62
63
64
65     System.out.println("Step4");
66
67
68
69     String name = "Reverse";
70
71     NameComponent path[] = ncRef.to_name(name);
72
73     ncRef.rebind(path,h_ref);
74
75
76
77     System.out.println("Reverse Server reading and waiting....");
78
79     orb.run();
80
81 }
82
83 catch (Exception e)
84
85 {
86
87     e.printStackTrace();
88
89 }
90
91 }
92
93 }
```