Reverse Server

```
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 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();
        }
    }
}
```

Reverse 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();
        }
    }
}
```

Reverse Implementation

```
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