```
// Adapted from Sun's tutorial
module HelloApp
{
   interface Hello
   {
      string sayHello();
      oneway void shutdown();
   };
};
```

```
import HelloApp.*;
import org.omg.CORBA.*;

public class HelloImpl extends HelloPOA
{
   private ORB orb;

   public void setORB(ORB orb_val)
   {
      orb = orb_val;
   }

   public String sayHello()
   {
      return "Hello world.";
   }

   public void shutdown()
   {
      orb.shutdown(false);
   }
}
```

```
import HelloApp.*;
import org.omg.CosNaming.*;
import org.omg.CosNaming.NamingContextPackage.*;
import org.omg.CORBA.*;
import org.omg.PortableServer.*;
import org.omg.PortableServer.POA;

public class HelloServer
{
   public static void main(String args[])
   {
      try {
         // where is the naming service?
         Properties props = new Properties();
         props.put("org.omg.CORBA.ORBInitialPort", "1050");
         props.put("org.omg.CORBA.ORBInitialHost", "falcon");

         // create and initialize the ORB
         ORB orb = ORB.init(args, props);

         // get reference to rootpoa & activate the POAManager
         POA rootpoa = POAHelper.narrow(orb.resolve_initial_references("RootPOA"));
         rootpoa.the_POAManager().activate();

         // create servant and register it with the ORB
         HelloImpl helloImpl = new HelloImpl();
         helloImpl.setORB(orb);

         // get object reference from the servant
         org.omg.CORBA.Object ref = rootpoa.servant_to_reference(helloImpl);
         Hello href = HelloHelper.narrow(ref);

         // get the root naming context
         org.omg.CORBA.Object objRef = orb.resolve_initial_references("NameService");

         // Use NamingContextExt which is part of the Interoperable
         // Naming Service (INS) specification.
         NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef);

         // bind the Object Reference in Naming
         NameComponent path[] = ncRef.to_name("Hello");
         ncRef.rebind(path, href);

         System.out.println("HelloServer ready and waiting ...");
         orb.run(); // wait for invocations from clients
      } catch (Exception e) {
         e.printStackTrace(System.out);
      }

      System.out.println("HelloServer Exiting ...");
   }
}
```

```
import HelloApp.*;
import org.omg.CosNaming.*;
import org.omg.CosNaming.NamingContextPackage.*;
import org.omg.CORBA.*;
import java.util.Properties;

public class HelloClient
{
   public static void main(String args[])
   {
      try
      {
         // where is the naming service?
         Properties props = new Properties();
         props.put("org.omg.CORBA.ORBInitialPort", "1050");
         props.put("org.omg.CORBA.ORBInitialHost", "falcon");

         // create and initialize the ORB
         ORB orb = ORB.init(args, props);

         // get the root naming context
         org.omg.CORBA.Object objRef =
         orb.resolve_initial_references("NameService");

         // Use NamingContextExt instead of NamingContext. This is
         // part of the Interoperable naming Service.
         NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef);

         // resolve the Object Reference in Naming
         Hello helloImpl = HelloHelper.narrow(ncRef.resolve_str("Hello"));

         System.out.println("Obtained a handle on server object: " + helloImpl);
         System.out.println(helloImpl.sayHello());
         helloImpl.shutdown();

      }
      catch (Exception e)
      {
         e.printStackTrace(System.out);
      }
   }
}
```