Program 1: Calculator

Calculator.idl

```
module CalculatorApp {
        interface Calculator {
        float add(in float a, in float b);
        float subtract(in float a, in float b);
        float multiply(in float a, in float b);
        float divide(in float a, in float b);
        };
};
```

CalculatorServer.java

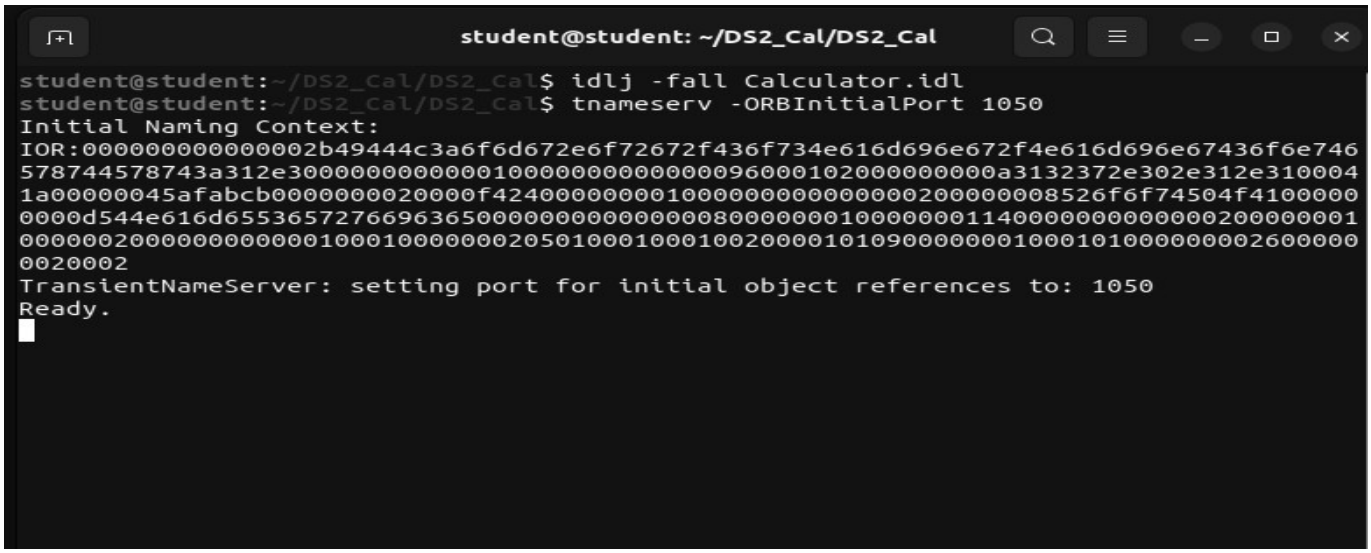
```
import CalculatorApp.*;
import CalculatorApp.CalculatorHelper;
import org.omg.CORBA.*;
import org.omg.PortableServer.*;
import org.omg.PortableServer.POA;
import CalculatorApp.*;
import CalculatorApp.CalculatorHelper;
import org.omg.CORBA.*;
import org.omg.CosNaming.*;
import org.omg.CosNaming.NamingContextExt;
import org.omg.CosNaming.NamingContextExtHelper;

class CalculatorImpl extends CalculatorPOA {
        private ORB orb;

        public void setORB(ORB orb) {
        this.orb = orb;
        }

        public float add(float a, float b) {
        return a + b;
        }

        public float subtract(float a, float b) {
        return a - b;
        }

        public float multiply(float a, float b) {
        return a * b;
        }
```

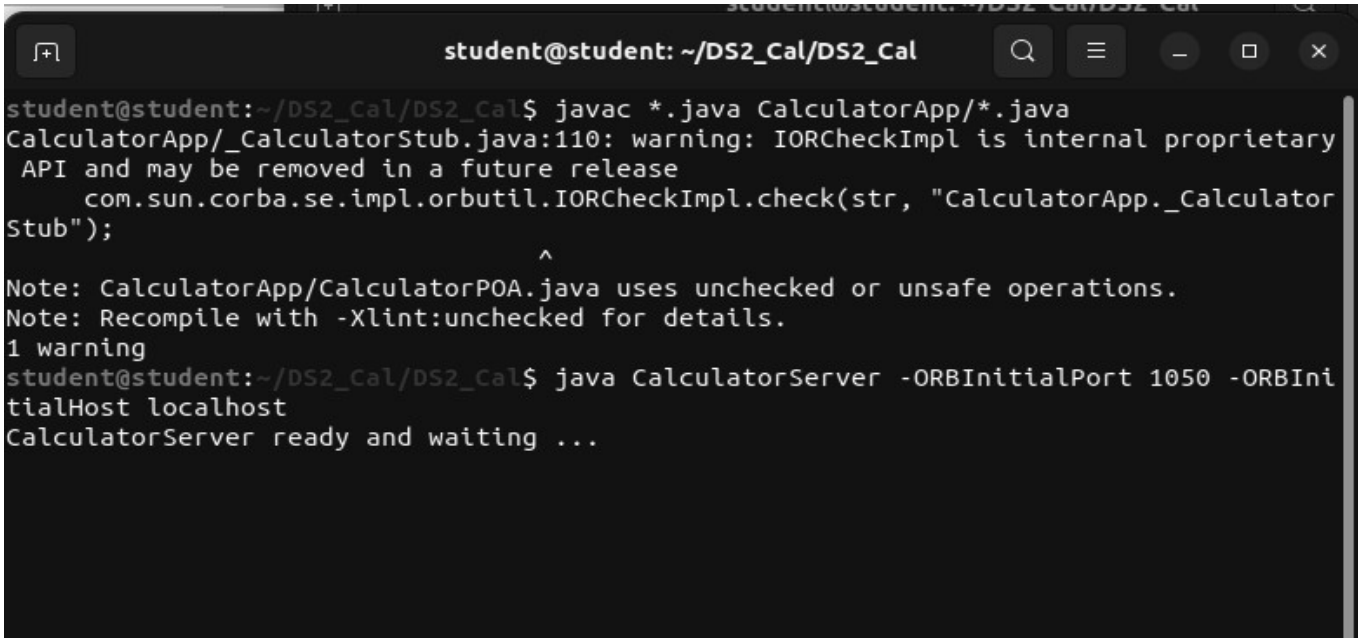
```
        public float divide(float a, float b) {
        if (b == 0) {
        throw new RuntimeException("Division by zero!");
        }
        return a / b;
        }
}

public class CalculatorServer {
        public static void main(String[] args) {
        try {
        // Initialize the ORB
        ORB orb = ORB.init(args, null);

        // Get reference to root POA and activate POA manager
        POA rootPoa = POAHelper.narrow(orb.resolve_initial_references("RootPOA"));
        rootPoa.the_POAManager().activate();

        // Create servant and register it with the ORB
        CalculatorImpl calculatorImpl = new CalculatorImpl();
        calculatorImpl.setORB(orb);

        // Get object reference from servant
        org.omg.CORBA.Object ref = rootPoa.servant_to_reference(calculatorImpl);
        Calculator href = CalculatorHelper.narrow(ref);

        // Bind the object reference in the naming service
        org.omg.CORBA.Object objRef = orb.resolve_initial_references("NameService");
        NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef);

        String name = "Calculator";
        NameComponent[] path = ncRef.to_name(name);
        ncRef.rebind(path, href);

        System.out.println("CalculatorServer ready and waiting ...");

        // Wait for incoming requests
        orb.run();
        } catch (Exception e) {
        e.printStackTrace();
        }
        }
}
```

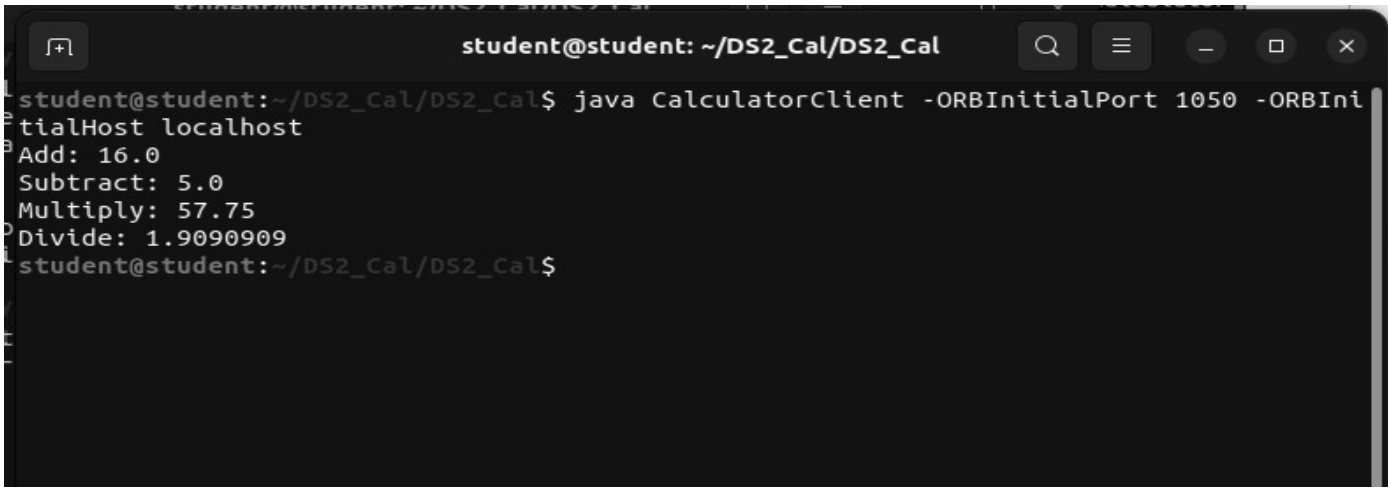
CalculatorClient.java

```
import CalculatorApp.*;
import CalculatorApp.CalculatorHelper;
import org.omg.CORBA.*;
import org.omg.CosNaming.*;
import org.omg.CosNaming.NamingContextExt;
import org.omg.CosNaming.NamingContextExtHelper;

public class CalculatorClient {
        public static void main(String[] args) {
        try {
        // Initialize ORB
        ORB orb = ORB.init(args, null);

        // Get reference to naming service
        org.omg.CORBA.Object objRef = orb.resolve_initial_references("NameService");
        NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef);

        // Resolve the Calculator object reference in the naming service
        String name = "Calculator";
        Calculator calculator = CalculatorHelper.narrow(ncRef.resolve_str(name));

        // Call remote methods
        System.out.println("Add: " + calculator.add(10.5f, 5.5f));
        System.out.println("Subtract: " + calculator.subtract(10.5f, 5.5f));
        System.out.println("Multiply: " + calculator.multiply(10.5f, 5.5f));
        System.out.println("Divide: " + calculator.divide(10.5f, 5.5f));
        } catch (Exception e) {
        e.printStackTrace();
        }
        }
}
```

Output:


student@student: ~/DS2_Cal/DS2_Cal
student@student:~/DS2_Cal/DS2_Cal$ idlj -fall Calculator.idl
student@student:~/DS2_Cal/DS2_Cal$ tnameserv -ORBInitialPort 1050
Initial Naming Context:
IOR:000000000000002b49444c3a6f6d672e6f72672f436f734e616d696e672f4e616d696e67436f6e746
578744578743a312e3000000000001000000000000009600010200000000a3132372e302e312e310004
1a00000045afabcb0000000020000f4240000000010000000000000002000000008526f6f74504f4100000
0000d544e616d65536572766963650000000000000008000000010000000114000000000000020000000100
0000020000000000001000100000002050100010001002000010109000000010001010000000026000000
0020002
TransientNameServer: setting port for initial object references to: 1050
Ready.


student@student: ~/DS2_Cal/DS2_Cal
student@student:~/DS2_Cal/DS2_Cal$ javac *.java CalculatorApp/*.java
CalculatorApp/_CalculatorStub.java:110: warning: IORCheckImpl is internal proprietary
API and may be removed in a future release
com.sun.corba.se.impl.orbutil.IORCheckImpl.check(str, "CalculatorApp._Calculator
Stub");
^
Note: CalculatorApp/CalculatorPOA.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
1 warning
student@student:~/DS2_Cal/DS2_Cal$ java CalculatorServer -ORBInitialPort 1050 -ORBIni
tialHost localhost
CalculatorServer ready and waiting ...


student@student: ~/DS2_Cal/DS2_Cal
student@student:~/DS2_Cal/DS2_Cal$ java CalculatorClient -ORBInitialPort 1050 -ORBIni
tialHost localhost
Add: 16.0
Subtract: 5.0
Multiply: 57.75
Divide: 1.9090909
student@student:~/DS2_Cal/DS2_Cal$

Program 2: String Operations

ReverseModule.idl

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

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));
   }
}
```

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 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();
        }
     }
}
```

ReverseClient.java

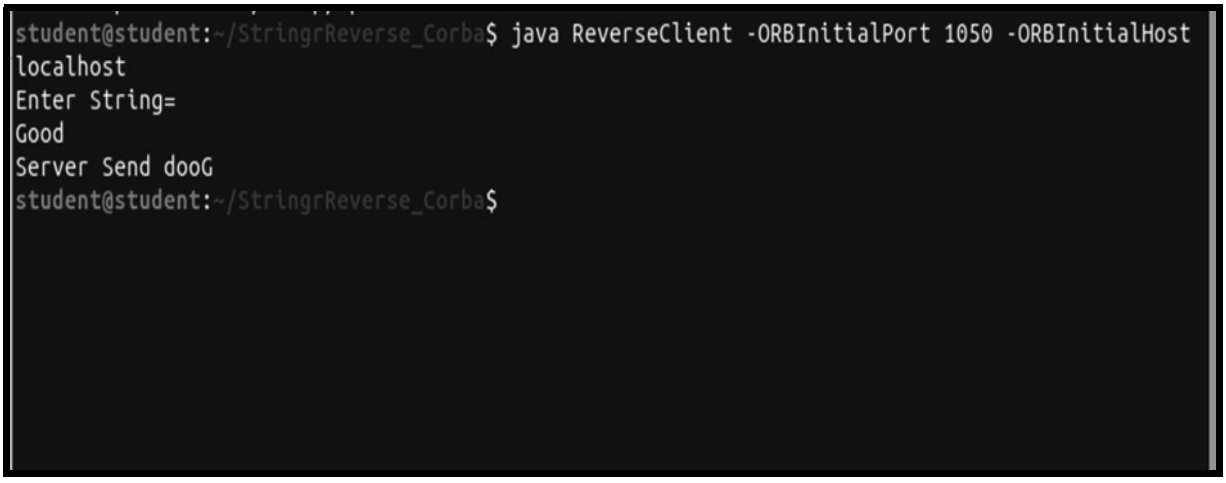
```
import ReverseModule.*;
```

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

Output:


student@student:~$ cd StringrReverse_Corba
student@student:~/StringrReverse_Corba$ idlj -fall ReverseModule.idl
student@student:~/StringrReverse_Corba$ 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
student@student:~/StringrReverse_Corba$ orbd -ORBInitialPort 1050&
[1] 7785
student@student:~/StringrReverse_Corba$


student@student:~/StringrReverse_Corba$ java ReverseServer -ORBInitialPort 1050& -ORBInitialHost localhost&
[1] 7824
[2] 7825
student@student:~/StringrReverse_Corba$ -ORBInitialHost: command not found
Reverse Object Created
Step1
Step2
Step3
Step4
Reverse Server reading and waiting....
student@student:~/StringrReverse_Corba$


student@student:~/StringrReverse_Corba$ java ReverseClient -ORBInitialPort 1050 -ORBInitialHost localhost
Enter String=
Good
Server Send dooG
student@student:~/StringrReverse_Corba$