**Name : Navnit Amrutharaj**

**Roll No: BIB02**

**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));

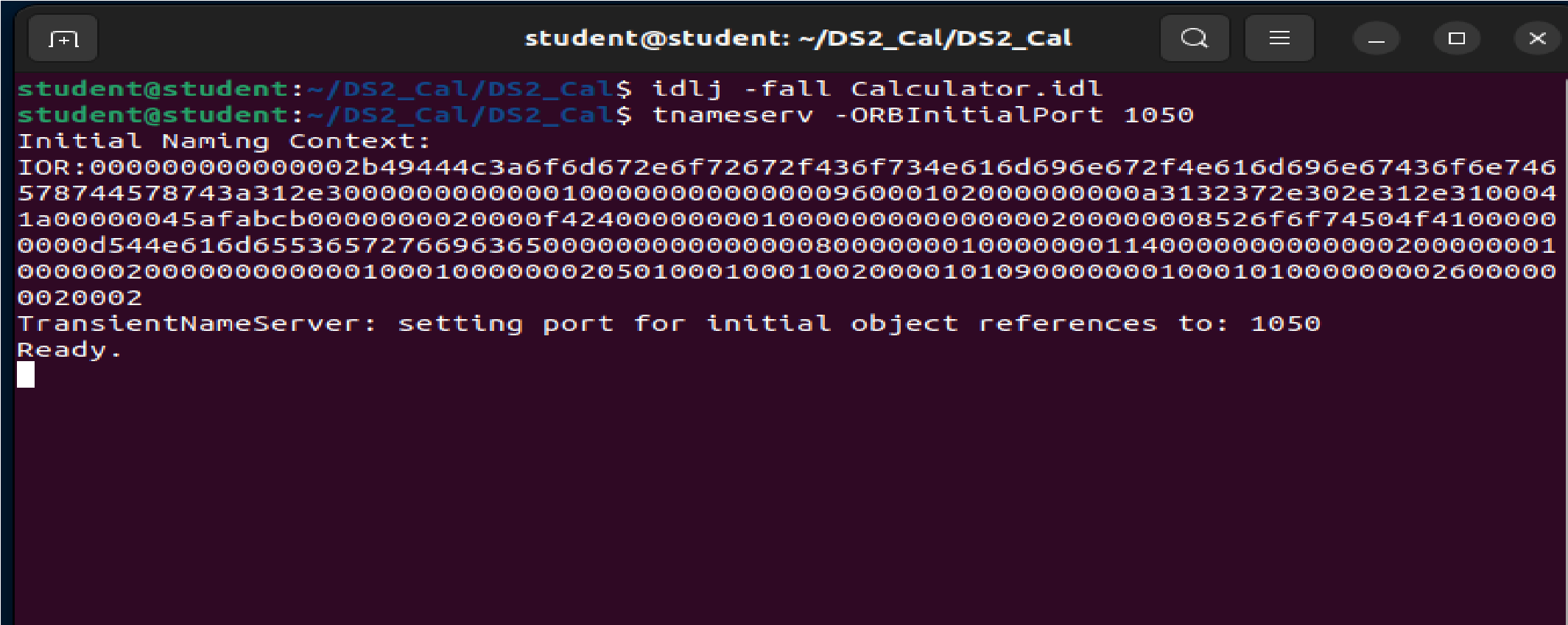
System.out.println("Divide: " + calculator.divide(10.5f, 5.5f)); } catch (Exception e) { e.printStackTrace();

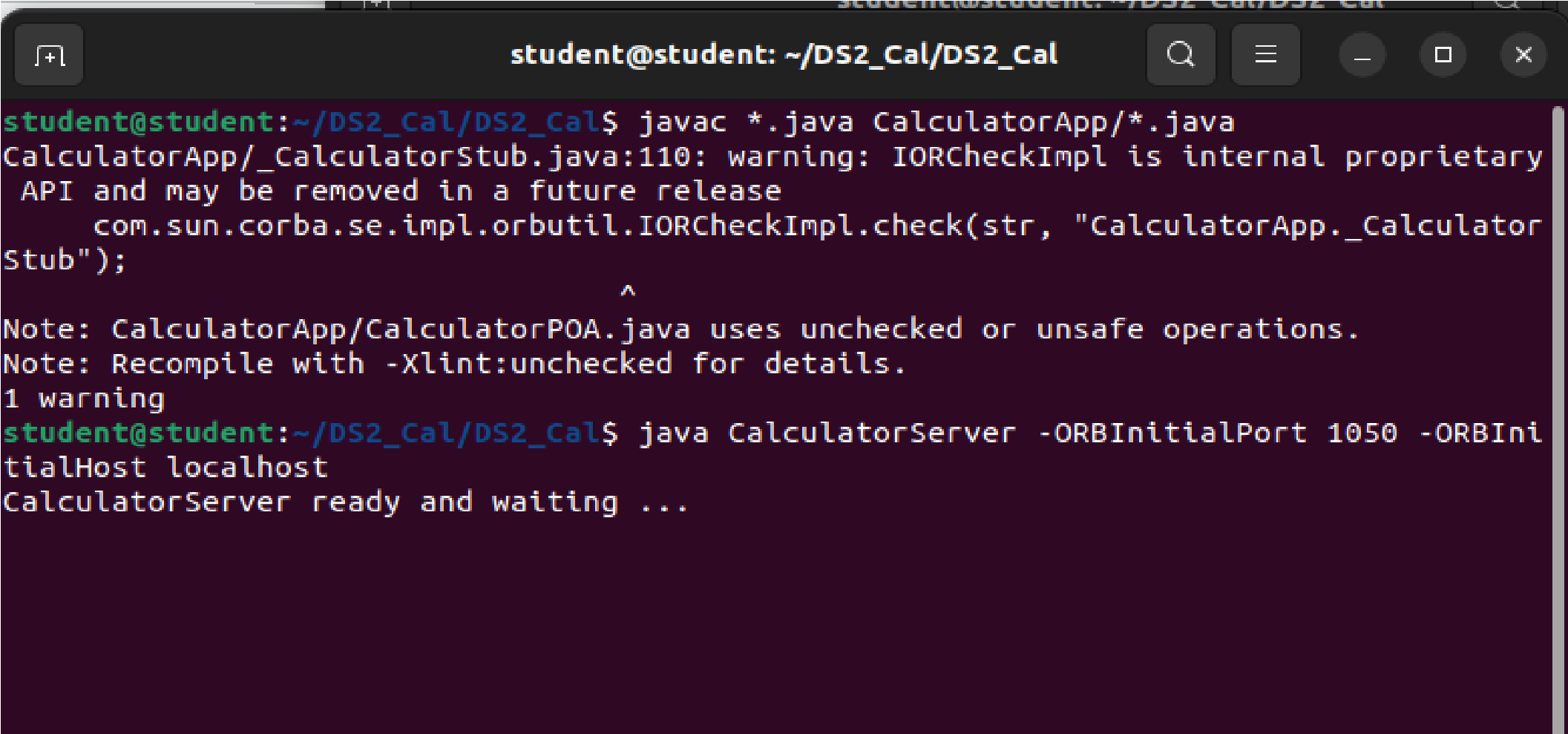
}

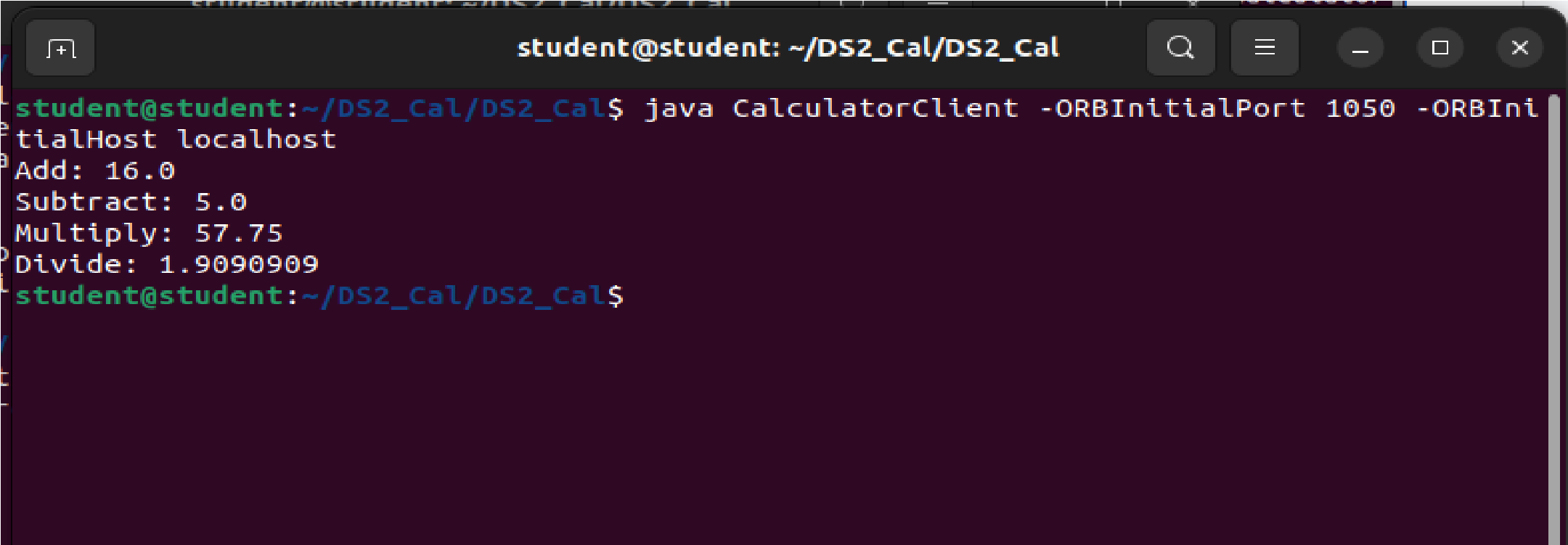
}

}

**Output:**







**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:**

