/*

CL-9
BE-9 P9
Roll No. 43104

**Assignment 3**

CORBA. Bank Agent: The application has Bank.idl file which contains an Account interface; provides a single method balance () for obtaining the current balance. The Account Manager interface creates an account for the user if one does not already exist. The Client class implements the client application which obtains the current
balance of a bank account. Server.java file implements the Server class for the server side of the banking application.

*/

**// calc.java**

```
module calc_val
{
 interface calc
 {
  double addfn(in double a,in double b);
  double subfn(in double a,in double b);
  double mulfn(in double a,in double b);
  double divfn(in double a,in double b);
 };
};
```

**// Client.java**

```
import calc_val.*;
import org.omg.CosNaming.*;
import org.omg.CosNaming.NamingContextPackage.*;
import org.omg.CORBA.*;
import java.io.*;
public class Client
{
  static calc calcimpl;
  public static void main(String args[])
```

```
{
 try
 {
  double result=0.0,num1=0.0,num2=0.0;
   // create and initialize the ORB
  ORB orb=ORB.init(args,null);
  // 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);
  String pathname="calc";
  calcimpl=calcHelper.narrow(ncref.resolve_str(pathname));
 int ch=1;
 while(ch!=0)
 {
  System.out.println("1. Addition");
  System.out.println("2. Subtraction");
  System.out.println("3. Multiplication");
  System.out.println("4. Division");
  System.out.println("0. Exit");
  BufferedReader in1=new BufferedReader(new InputStreamReader(System.in));
  System.out.println("enter your choice: ");
  ch=Integer.parseInt(in1.readLine());

  if(ch==0)
    break;
  BufferedReader in=new BufferedReader(new InputStreamReader(System.in));
  System.out.println("enter number1: ");
  num1=Double.parseDouble(in.readLine());
  System.out.println("enter number2: ");
  num2=Double.parseDouble(in.readLine());
  switch(ch)
```

```
      {
        case 1:
           result=calcimpl.addfn(num1,num2);
           break;
        case 2:
           result=calcimpl.subfn(num1,num2);
           break;
        case 3:
           result=calcimpl.mulfn(num1,num2);
           break;
        case 4:
           result=calcimpl.divfn(num1,num2);
           break;
      }
           System.out.println("result is: "+result);
      }
    }
    catch(Exception e)
    {
      System.out.println(e);
    }
  }
}
```

**// Server.java**

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

class serverimpl extends calcPOA
{
  private ORB orb;
  public void setorb(ORB orb_val)
  {   orb=orb_val;   }
  public double addfn(double n1,double n2)
  {
    return(n1+n2);
  }
  public double subfn(double n1,double n2)
  {
    return(n1-n2);
  }
  public double mulfn(double n1,double n2)
  {
    return(n1*n2);
```

```
  }
  public double divfn(double n1,double n2)
  {
   return(n1/n2);
  }
}
public class Server
{
  public static void main(String args[])
  {
   try
   {
      //Create and initialize the ORB
      ORB orb=ORB.init(args,null);

       // get reference to rootpoa & activate the POAManager
      org.omg.CORBA.Object objref1=orb.resolve_initial_references("RootPOA");
      POA rootpoa=POAHelper.narrow(objref1);
      rootpoa.the_POAManager().activate();
       //create servant and register it with the ORB
      serverimpl serverobj=new serverimpl();
      serverobj.setorb(orb);
       // get object reference from the servant
      org.omg.CORBA.Object objref2=rootpoa.servant_to_reference(serverobj);
      calc href=calcHelper.narrow(objref2);
      // get the root naming context
      org.omg.CORBA.Object objref3= orb.resolve_initial_references("NameService");

       // Use NamingContextExt which is part of the Interoperable Naming Service (INS) specification.
      NamingContextExt ncref=NamingContextExtHelper.narrow(objref3);
      String pathname="calc";
      NameComponent path[]=ncref.to_name(pathname);
```

```
        // bind the Object Reference in Naming
        ncref.rebind(path,href);
        System.out.println("server ready and waiting...");
        // wait for invocations from clients
        orb.run();
      }
      catch(Exception e)
      {
        System.out.println(e);
      }
    }
}
```

## // OUTPUT

**//Server**

Ajinkya@Tikhe-Inspiron-7559:~/Aj/CL-9/assignment_3$ javac *.java

Ajinkya@Tikhe -Inspiron-7559:~/Aj/CL-9/assignment_3$ Start orbd -ORBInitialPort 1050&

Ajinkya@Tikhe -Inspiron-7559:~/Aj/CL-9/assignment_3$ java calc server -ORBInitialPort 1050 -ORBInitialHost localhost&

server ready and waiting...

**// Client**

Ajinkya@Tikhe -Inspiron-7559:~/Aj/CL-9/assignment_3$ java calc_client -ORBInitialPort 1050 -ORBInitialHost localhost

1. Addition

2. Subtraction

3. Multiplication

4. Division

0. Exit

enter your choice:

1

enter number1:

10

enter number2:

10

result is: 20.0

1. Addition

2. Subtraction

3. Multiplication

4. Division

0. Exit

enter your choice:

2

enter number1:

20

enter number2:

10

result is: 10.0

1. Addition

2. Subtraction

3. Multiplication

4. Division

0. Exit

enter your choice:

0