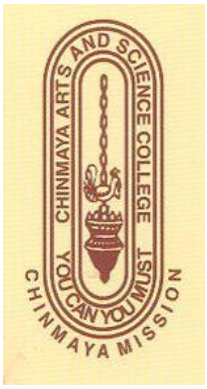
CHINMAYA ARTS AND SCIENCE COLLEGE FOR WOMEN

GOVINDAGIRI, CHALA, KANNUR



SIXTH SEMESTER BCA

PRACTICAL RECORD

2023-24

on

Enterprise Java Programming

CHINMAYA ARTS AND SCIENCE COLLEGE FOR WOMEN

GOVINDAGIRI, CHALA, KANNUR

CERTIFICATE

It is certified that this is a bonafide record of the original work done by Ms ..... Reg.No ..... of Sixth Semester BCA on 6B21BCA Lab V: Enterprise Java Programming during the year 2023-24.

Principal : ***Faculty in charge : Ms Raji K***

External Examiner :

1.

2.

Submitted for practical examination held on .....

INDEX

ENTERPRISE JAVA PROGRAMMING

1. INSERT, UPDATE AND DELETE RECORD INTO EMPLOYEE TABLE

AIM: Write a JDBC program to insert, update and delete record into employee table.

PROGRAM:

File:employee.java

```
import java.sql.*;
import java.io.*;
public class employee
{
	public static void main(String args[])throws IOException
	{
		int ch,upc;
		int no,sal;
		String name;
		BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
		try
		{
			Class.forName("org.postgresql.Driver");
		}
		catch(ClassNotFoundException e)
		{
			System.out.println("Unable to load driver:"+e);
		}
		try
		{
			Connection
con=DriverManager.getConnection("jdbc:postgresql://localhost/bca3","bca3","redhat");
			Statement stmt=con.createStatement();
			System.out.println("\n.....Current Records.....\n");
			System.out.println("ENO\tENAME\tSALARY\n");
			ResultSet rs=stmt.executeQuery("select * from employee");
			while(rs.next())
			{

System.out.println(rs.getInt("eno")+"\t"+rs.getString("ename")+"\t"+rs.getInt("salary"));
			}
			do
			{

System.out.println("\nMENU\n1.Insert\n2.Update\n3.Delete\n4.Display\n5.Exit\nEnter your
choice:");
				ch=Integer.parseInt(br.readLine());
				switch(ch)
				{
```

```
                        case 1:System.out.println("Enter employee number,name and salary:");
                            no=Integer.parseInt(br.readLine());
                            name=br.readLine();
                            sal=Integer.parseInt(br.readLine());
                            stmt.executeUpdate("insert into employee values("+no+",' "+name+"
','"+sal+");");
                            System.out.println("Record inserted");
                            break;
                        case 2:System.out.println("Enter employee number of the record to be
updated:");
                            no=Integer.parseInt(br.readLine());
                            System.out.println("Enter new name and salary:");
                            name=br.readLine();
                            sal=Integer.parseInt(br.readLine());
                            upc=stmt.executeUpdate("update employee set ename=' "+name+"
',salary="+sal+"where eno="+no);
                            if(upc!=0)
                                System.out.println("Record updated");
                            else
                                System.out.println("No such record exist");
                            break;
                        case 3:System.out.println("Enter employee number of the record you want
to delete:");
                            no=Integer.parseInt(br.readLine());
                            upc=stmt.executeUpdate("delete from employee where eno="+no+";");
                            if(upc!=0)
                                System.out.println("Record deleted");
                            else
                                System.out.println("No such record exist");
                            break;
                        case 4:ResultSet rs1=stmt.executeQuery("select * from employee");
                            System.out.println("ENO\tENAME\tSALARY\n");
                            while(rs1.next())
                            {

System.out.println(rs1.getInt("eno")+"\t"+rs1.getString("ename")+"\t"+rs1.getInt("salary"));
                            }
                            break;
                        case 5:System.exit(0);
                        default:System.out.println("Enter a valid choice");
                    }
                }while(ch!=5);
                rs.close();
                stmt.close();
                con.close();
            }
            catch(SQLException e)
            {
                System.out.println("Connection failed:"+e.getMessage());
            }
```

```
    }
}
```

Compile and Execute:

```
bca3@chinarts-Veriton-M200-H410:~/Desktop$ javac employee.java
bca3@chinarts-Veriton-M200-H410:~/Desktop$ locate *.jar
.....
/usr/share/java/postgresql-jdbc3-9.1.jar
.....
bca3@chinarts-Veriton-M200-H410:~/Desktop$ export
CLASSPATH=$CLASSPATH:/usr/share/java/postgresql-jdbc3-9.1.jar
bca3@chinarts-Veriton-M200-H410:~/Desktop$ java employee
```

OUTPUT:

```
.....Current Records.....

ENO   ENAME      SALARY

1     Ayan       50000
2     Anami      150000
3     Aman       15000
4     Rachna     100000
5     Samiya     1500000

MENU
1.Insert
2.Update
3.Delete
4.Display
5.Exit
Enter your choice:
1
Enter employee number,name and salary:
6
Durga
100000
Record inserted

MENU
1.Insert
2.Update
3.Delete
4.Display
5.Exit
Enter your choice:
4
ENO   ENAME      SALARY

1     Ayan       50000
2     Anami      150000
3     Aman       15000
4     Rachna     100000
5     Samiya     1500000
6     Durga      100000

MENU
1.Insert
2.Update
3.Delete
4.Display
5.Exit
Enter your choice:
```

```
2
Enter employee number of the record to be updated:
6
Enter new name and salary:
Durga.Jai
150000
Record updated

MENU
1.Insert
2.Update
3.Delete
4.Display
5.Exit
Enter your choice:
4
ENO   ENAME       SALARY

1       Ayan        50000
2       Anami       150000
3       Aman        15000
4       Rachna      100000
5       Samiya      1500000
6       Durga.Jai   150000

MENU
1.Insert
2.Update
3.Delete
4.Display
5.Exit
Enter your choice:
3
Enter employee number of the record you want to delete:
6
Record deleted

MENU
1.Insert
2.Update
3.Delete
4.Display
5.Exit
Enter your choice:
4
ENO   ENAME       SALARY

1       Ayan        50000
2       Anami       150000
3       Aman        15000
```

```
4       Rachna      100000
5       Samiya      1500000

MENU
1.Insert
2.Update
3.Delete
4.Display
5.Exit
Enter your choice:
7
Enter a valid choice

MENU
1.Insert
2.Update
3.Delete
4.Display
5.Exit
Enter your choice:
5
```

2. SCROLLING FUNCTION

AIM: Write a JDBC Program to connect Student table .Implement the record scrolling functions-first() , last() , next() , previous() , absolute() , and relative().

PROGRAM:

File:student.java

```
import java.sql.*;
import java.io.*;
public class student
{
	public static void main(String args[])throws IOException
	{
		BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
		int ch;
		try
		{
			Class.forName("org.postgresql.Driver");
		}
		catch(ClassNotFoundException e)
		{
			System.out.println("Unable to load class:"+e);
		}
		try
		{
			Connection
con=DriverManager.getConnection("jdbc:postgresql://localhost/bca3","bca3","redhat");
			Statement stmt=con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE);
			System.out.println("\nRNO\tNAME\tMARK\n");
			ResultSet rs=stmt.executeQuery("select * from student");
			while(rs.next())
			{

System.out.println(rs.getInt("rno")+"\t"+rs.getString("sname")+"\t"+rs.getInt("mark"));
			}
			rs.first();
			do
			{
				System.out.println("\n.....MENU.....\n1.Move to first row\n2.Move to next
row\n3.Move to previous row\n4.Move to particular row\n5.Move to relative row\n6.Move to last
row\n7.Exit\n\nEnter your choice:");
				ch=Integer.parseInt(br.readLine());
				switch(ch)
				{
					case 1:System.out.println("\nMove to first row:");
						rs.first();
```

```
System.out.println(rs.getInt("rno")+"\t"+rs.getString("sname")+"\t"+rs.getInt("mark"));
                                    break;
                                case 2:if(rs.isLast())
                                            System.out.println("Already in last row");
                                    else
                                    {
                                            System.out.println("\nMove to next row:");
                                            rs.next();

System.out.println(rs.getInt("rno")+"\t"+rs.getString("sname")+"\t"+rs.getInt("mark"));
                                    }
                                    break;
                                case 3:if(rs.isFirst())
                                            System.out.println("Already in first row");
                                    else
                                    {
                                            System.out.println("\nMove to previous row:");
                                            rs.previous();

System.out.println(rs.getInt("rno")+"\t"+rs.getString("sname")+"\t"+rs.getInt("mark"));
                                    }
                                    break;
                                case 4:System.out.println("Move to a paricular row:");
                                    System.out.println("Enter the row number:");
                                    int no=Integer.parseInt(br.readLine());
                                    try
                                    {
                                            rs.absolute(no);

System.out.println(rs.getInt("rno")+"\t"+rs.getString("sname")+"\t"+rs.getInt("mark"));
                                    }
                                    catch(SQLException e)
                                    {
                                            System.out.println("Record does not exist");
                                            rs.first();
                                    }
                                    break;
                                case 5:System.out.println("Move to a relative row:");
                                    System.out.println("Enter the row number:");
                                    int n1=Integer.parseInt(br.readLine());
                                    try
                                    {
                                            rs.relative(n1);

System.out.println(rs.getInt("rno")+"\t"+rs.getString("sname")+"\t"+rs.getInt("mark"));
                                    }
                                    catch(SQLException e)
                                    {
                                            System.out.println("Record does not exist");
```

```
                                    rs.first();
                                }
                                break;
                            case 6:System.out.println("Move to a last row:");
                                rs.last();

System.out.println(rs.getInt("rno")+"\t"+rs.getString("sname")+"\t"+rs.getInt("mark"));
                                break;
                            case 7:System.exit(0);
                                break;
                            default:System.out.println("Invalid choice");
                                 break;
                        }
                    }while(ch!=7);
                    rs.close();
                    stmt.close();
                    con.close();
                }
                catch(SQLException e)
                {
                    System.out.println(e);
                }
        }
}
```

Compile and Execute:

```
bca3@chinarts-Veriton-M200-H410:~$ javac student.java
bca3@chinarts-Veriton-M200-H410:~$ locate *.jar
/home/bca3/postgresql-jdbc3-9.1.jar
/usr/share/java/postgresql-jdbc3-9.1.jar
bca3@chinarts-Veriton-M200-H410:~$ export
CLASSPATH=$CLASSPATH:/usr/share/java/postgresql-jdbc3-9.1.jar
bca3@chinarts-Veriton-M200-H410:~$ java student
```

OUTPUT:

```
RNO   NAME   MARK

1       Anami    50
2       Bill       35
3       Coups    50
4       Durga    49
5       Emil      20

......MENU......
1.Move to first row
2.Move to next row
3.Move to previous row
4.Move to particular row
5.Move to relative row
6.Move to last row
7.Exit

Enter your choice:
1

Move to first row:
1       Anami    50

......MENU......
1.Move to first row
2.Move to next row
3.Move to previous row
4.Move to particular row
5.Move to relative row
6.Move to last row
7.Exit

Enter your choice:
2

Move to next row:
2       Bill       35

......MENU......
1.Move to first row
2.Move to next row
3.Move to previous row
4.Move to particular row
5.Move to relative row
6.Move to last row
7.Exit

Enter your choice:
```

```
3

Move to previous row:
1        Anami     50

......MENU......
1.Move to first row
2.Move to next row
3.Move to previous row
4.Move to particular row
5.Move to relative row
6.Move to last row
7.Exit

Enter your choice:
4
Move to a paricular row:
Enter the row number:
3
3        Coups     50

......MENU......
1.Move to first row
2.Move to next row
3.Move to previous row
4.Move to particular row
5.Move to relative row
6.Move to last row
7.Exit

Enter your choice:
5
Move to a relative row:
Enter the row number:
1
4        Durga     49

......MENU......
1.Move to first row
2.Move to next row
3.Move to previous row
4.Move to particular row
5.Move to relative row
6.Move to last row
7.Exit

Enter your choice:
6
Move to a last row:
5        Emil       20
```

```
.....MENU.....
1.Move to first row
2.Move to next row
3.Move to previous row
4.Move to particular row
5.Move to relative row
6.Move to last row
7.Exit

Enter your choice:
7
```

3. DATABASE METADATA

AIM: Write a JDBC program to display database metadata.

PROGRAM:

File:DBViewer.java

```
import java.sql.*;
import java.util.StringTokenizer;
public class DBViewer
{
	final static String jdbcURL="jdbc:postgresql://localhost/bca3";
	final static String jdbcDriver="org.postgresql.Driver";
	public static void main(String args[])
	{
		System.out.println(".....DATABASE VIEWER.....\n");
		try
		{
			Class.forName(jdbcDriver);
			Connection con=DriverManager.getConnection(jdbcURL,"bca3","redhat");
			DatabaseMetaData dbmd=con.getMetaData();
			System.out.println("\nDriver Name="+dbmd.getDriverName());
			System.out.println("Database Product name="+dbmd.getDatabaseProductName());
			System.out.println("SQLkeyword supported:\n");
			StringTokenizer st=new StringTokenizer(dbmd.getSQLKeywords(),",");
			while(st.hasMoreTokens())
				System.out.print(st.nextToken()+"  ");
			String[] tabletypes={"TABLE"};
			ResultSet allTables=dbmd.getTables(null,null,null,tabletypes);
			while(allTables.next())
			{
				String table_name=allTables.getString("TABLE_NAME");
				System.out.println("\nTable name:"+table_name);
				System.out.println("Table type="+allTables.getString("TABLE_TYPE"));
			}
			allTables.close();
			con.close();
		}
		catch(ClassNotFoundException e)
		{
			System.out.println("Unable to load driver class"+e);
		}
		catch(SQLException e)
		{
			System.out.println("SQLException"+e.getMessage());
		}
	}
}
```

Compile and Execute:

```
bca3@chinarts-Veriton-M200-H410:~$ javac DBViewer.java
bca3@chinarts-Veriton-M200-H410:~$ locate *.jar
/home/bca3/postgresql-jdbc3-9.1.jar
/usr/share/java/postgresql-jdbc3-9.1.jar
bca3@chinarts-Veriton-M200-H410:~$ export
CLASSPATH=$CLASSPATH:/usr/share/java/postgresql-jdbc3-9.1.jar
bca3@chinarts-Veriton-M200-H410:~$ java DBViewer
```

OUTPUT:

.....DATABASE VIEWER.....

Driver Name=PostgreSQL Native Driver
Database Product name=PostgreSQL
SQLkeyword supported:

abort acl add aggregate append archive arch_store backward binary boolean change cluster copy database delimiter delimiters do extend explain forward heavy index inherits isnull light listen load merge nothing notify notnull oids purge rename replace retrieve returns rule recipe setof stdin stdout store vacuum verbose version
Table name:customer
Table type=TABLE

Table name:customerz
Table type=TABLE

Table name:employ
Table type=TABLE

Table name:employee
Table type=TABLE

Table name:employee1
Table type=TABLE

Table name:employeee
Table type=TABLE

Table name:emply
Table type=TABLE

Table name:sql_features
Table type=TABLE

Table name:sql_implementation_info
Table type=TABLE

Table name:sql_languages
Table type=TABLE

Table name:sql_packages
Table type=TABLE

Table name:sql_parts
Table type=TABLE

Table name:sql_sizing

Table type=TABLE

Table name:sql_sizing_profiles
Table type=TABLE

Table name:student
Table type=TABLE

Table name:students
Table type=TABLE

4. RESULTSET METADATA

AIM: Write a JDBC program to display ResultSet metadata.

PROGRAM:

File:TableViewer.java

```
import java.sql.*;
public class TableViewer
{
	final static String jdbcURL="jdbc:postgresql://localhost/bca3";
	final static String jdbcDriver="org.postgresql.Driver";
	final static String table="customerz";
	public static void main(String args[])
	{
		System.out.println(".....TABLE VIEWER.....\n");
		try
		{
			Class.forName(jdbcDriver);
			Connection con=DriverManager.getConnection(jdbcURL,"bca3","redhat");
			Statement stmt=con.createStatement();
			ResultSet rs=stmt.executeQuery("select * from "+table);
			ResultSetMetaData rsmd=rs.getMetaData();
			int ColumnCount=rsmd.getColumnCount();
			for(int col=1;col<=ColumnCount;col++)
			{
				System.out.print(rsmd.getColumnLabel(col));
				System.out.print("("+rsmd.getColumnTypeName(col)+")");
				if(col<ColumnCount)
					System.out.print("\t");
			}
			System.out.println();
			while(rs.next())
			{
				for(int col=1;col<=ColumnCount;col++)
				{
					System.out.print(rs.getString(col)+"\t");
					if(col<ColumnCount)
						System.out.print("\t");
				}
				System.out.println();
			}
			rs.close();
			stmt.close();
			con.close();
		}
		catch(ClassNotFoundException e)
		{
```

```
                System.out.println("Unable to load driver class");
            }
            catch(SQLException e)
            {
                System.out.println("SQLException:"+e.getMessage());
            }
        }
}
```

Compile and Execute:

```
bca3@chinarts-Veriton-M200-H410:~$ javac TableViewer.java
bca3@chinarts-Veriton-M200-H410:~$ export
CLASSPATH=$CLASSPATH:/usr/share/java/postgresql-jdbc3-9.1.jar
bca3@chinarts-Veriton-M200-H410:~$ java TableViewer
```

OUTPUT:

.....TABLE VIEWER.....

cid(int4)	cname(bpchar)
1	ANAMI
2	AMAN
3	RIYA

5. RMI COMPLEX NUMBER OPERATION

AIM: Write a RMI program to perform complex number operations.

PROGRAM:

File:complex1.java

```
import java.io.Serializable;
public class complex1 implements Serializable
{
	public int r,i;
	public complex1()
	{
		r=0;
		i=0;
	}
	public complex1(int x,int y)
	{
		r=x;
		i=y;
	}
}
```

Compile:
bca3@chinarts-Veriton-M200-H410:~$javac complex1.java

File:vi complexInter.java

```
import java.rmi.Remote;
import java.rmi.RemoteException;
public interface complexInter extends Remote
{
	public complex1 add(complex1 x,complex1 y)throws RemoteException;
	public complex1 sub(complex1 x,complex1 y)throws RemoteException;
	public complex1 mul(complex1 x,complex1 y)throws RemoteException;
}
```

Compile:
bca3@chinarts-Veriton-M200-H410:~$javac complexInter.java

File:vi complexImpl.java

```
import java.rmi.server.UnicastRemoteObject;
import java.rmi.RemoteException;
public class complexImpl extends UnicastRemoteObject implements complexInter
{
	public complex1 c;
	public complexImpl(int a,int b)throws RemoteException
```

```
    {
        c=new complex1(a,b);
    }
    public complexImpl()throws RemoteException{}
    public complex1 add(complex1 x,complex1 y)throws RemoteException
    {
        complex1 z=new complex1();
        z.r=x.r+y.r;
        z.i=x.i+y.i;
        return z;
    }
    public complex1 sub(complex1 x,complex1 y)throws RemoteException
    {
        complex1 z=new complex1();
        z.r=x.r-y.r;
        z.i=x.i-y.i;
        return z;
    }
    public complex1 mul(complex1 x,complex1 y)throws RemoteException
    {
        complex1 z=new complex1();
        z.r=((x.r)*(y.r))-((x.i)*(y.i));
        z.i=((x.r)*(y.i))+((y.r)*(x.i));
        return z;
    }
}
```

Compile and Execute:

```
bca3@chinarts-Veriton-M200-H410:~$javac complexImpl.java
bca3@chinarts-Veriton-M200-H410:~$pwd
/home/bca3/EJP
bca3@chinarts-Veriton-M200-H410:~$exportCLASSPATH=$CLASSPATH:/home/bca3/EJP
bca3@chinarts-Veriton-M200-H410:~$rmic complexImpl
```

File:vi complexReg.java

```
import java.rmi.Naming;
public class complexReg
{
    public static void main(String args[])
    {
        try
        {
            complexImpl comp=new complexImpl();
            Naming.rebind("com",comp);
            System.out.println("Object Registered");
        }
        catch(Exception e)
        {
```

```
                e.printStackTrace();
            }
        }
}
```

Compile and Execute:

```
bca3@chinarts-Veriton-M200-H410:~$javac complexReg.java
bca3@chinarts-Veriton-M200-H410:~$rm1registry &
[1] 3984
bca3@chinarts-Veriton-M200-H410:~$java complexReg
Object Registered
```

File:vi complexClient

```
import java.rmi.Naming;
import java.io.*;
public class complexClient
{
    public static void main(String args[])throws IOException
    {
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        int x1,y1,x2,y2;
        int ch;
        System.out.println("Enter the real and imaginary part of first complex number:");
        x1=Integer.parseInt(br.readLine());
        y1=Integer.parseInt(br.readLine());
        System.out.println("Enter the real and imaginary part of second complex number:");
        x2=Integer.parseInt(br.readLine());
        y2=Integer.parseInt(br.readLine());
        complex1 a1=new complex1(x1,y1);
        complex1 a2=new complex1(x2,y2);
        complex1 a3;
        try
        {
            complexInter obj=(complexInter)Naming.lookup("rmi://localhost/com");
            System.out.println("Complex numbers are:");
            System.out.println(a1.r+"+"+a1.i+"i");
            System.out.println(a2.r+"+"+a2.i+"i");
            do
            {
                System.out.println("\nMENU\n1.Add\n2.Subtracr\n3.Multiply\n4.Exit\n\nEnter
your choice:");
                ch=Integer.parseInt(br.readLine());
                switch(ch)
                {
                    case 1:a3=obj.add(a1,a2);
                        System.out.println("Sum:"+a3.r+"+"+a3.i+"i");
                        break;
                    case 2:a3=obj.sub(a1,a2);
                        System.out.println("Difference:"+a3.r+"+"+a3.i+"i");
```

```
                                break;
                            case 3:a3=obj.mul(a1,a2);
                                System.out.println("Difference:"+a3.r+"+"+a3.i+"i");
                                break;
                            case 4:System.exit(0);
                            default:System.out.println("Enter a valid choice");
                        }
                    }while(ch!=4);
                }
                catch(Exception e)
                {
                    System.out.println("Error:"+e);
                }
        }
}
```

Compile and Execute:
bca3@chinarts-Veriton-M200-H410:~$javac complexClient.java
bca3@chinarts-Veriton-M200-H410:~$java complexClient

OUTPUT:

```
Enter the real and imaginary part of first complex number:
6
4
Enter the real and imaginary part of second complex number:
4
2
Complex numbers are:
6+4i
4+2i

MENU
1.Add
2.Subtracr
3.Multiply
4.Exit

Enter your choice:
1
Sum:10+6i

MENU
1.Add
2.Subtracr
3.Multiply
4.Exit

Enter your choice:
2
Difference:2+2i

MENU
1.Add
2.Subtracr
3.Multiply
4.Exit

Enter your choice:
3
Difference:16+28i

MENU
1.Add
2.Subtracr
3.Multiply
4.Exit

Enter your choice:
4
```

6. RMI BANK OPERATION

AIM: Write a RMI program to perform bank operation.

PROGRAM:

File:Account.java

```
import java.rmi.Remote;
import java.rmi.RemoteException;
public interface Account extends Remote
{
        public String getName() throws RemoteException;
        public float getBalance() throws RemoteException;
        public int  withdraw(float amt) throws RemoteException;
        public void deposit(float amt) throws RemoteException;
}
```

Compile:
bca3@chinarts-H410MH:~/Desktop/EJP$ javac Account.java

File:AccountImpl.java

```
import java.rmi.server.UnicastRemoteObject;
import java.rmi.RemoteException;
public class AccountImpl extends UnicastRemoteObject implements Account
{
    private float balance=0;
    private String name="";

    public AccountImpl(String name1)throws RemoteException
    {
        name=name1;
    }
    public String getName() throws RemoteException
    {
        return name;
    }
    public float getBalance() throws RemoteException
    {
        return balance;
    }
    public int withdraw(float amt) throws RemoteException
    {
        if((balance-amt)>1000)
        {
            balance-=amt;
            return 0;
        }
```

```
            else return -1;
      }
      public void  deposit(float amt) throws RemoteException
      {
            balance+=amt;

      }
}
```

Compile:
bca3@chinarts-H410MH:~/Desktop/EJP$ javac AccountImpl.java

File:RegAccount.java

```
import java.rmi.Naming;
public class RegAccount
{
      public static void main(String args[])
      {
            try
            {
                  Account acct=new AccountImpl(args[0]);
                  Naming.rebind(args[0],acct);
                  System.out.println("Registered Account");
            }
            catch (Exception e)
            {
                  e.printStackTrace();
            }
      }
}
```

Compile and Execute:
bca3@chinarts-Veriton-M200-H410:~/Desktop$ javac AccountImpl.java
bca3@chinarts-Veriton-M200-H410:~/Desktop$ java RegAccount Jimin
Registered Account

File:AccountClient.java

```
import java.io.*;
import java.rmi.Naming;
public class AccountClient
{
      public static void main(String args[])throws IOException
      {
            String name;
            int ch;
            float amount,bal,flag;
            try
            {
```

```
                Account obj=(Account)Naming.lookup("rmi://localhost/"+args[0]);
                BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
                do
                {
                    System.out.print("\nMENU\n1.Deposit\n2.Withdraw\n3.Display\n4.Exit\nEnter
your choice:");
                    ch=Integer.parseInt(br.readLine());
                    switch(ch)
                    {
                        case 1:System.out.println("Enter the amount to be deposited:");
                            amount=Float.parseFloat(br.readLine());
                            obj.deposit(amount);
                            System.out.println("Deposited Successfully");
                            System.out.println("New Balance:"+obj.getBalance());
                            break;
                        case 2:System.out.println("Enter the amount to be withdrawn:");
                            amount=Float.parseFloat(br.readLine());
                            flag=obj.withdraw(amount);
                            if(flag!=-1)
                            {
                                bal=obj.getBalance();
                                System.out.println("Withdrawn Successfully\nNew Balance:
"+bal);
                            }
                            else
                                System.out.println("No Sufficient Balance");
                            break;
                        case 3:System.out.println("Name:"+obj.getName());
                            System.out.println("Current Balance:"+obj.getBalance());
                            break;
                        case 4:System.exit(0);
                        default:System.out.println("Enter a avalid choice:");
                    }
                }while(ch!=4);
            }
            catch(Exception e)
            {
                System.out.print("Error"+e);
            }
        }
}
```

Compile and Execute:
bca3@chinarts-Veriton-M200-H410:~/Desktop$ javac AccountClient.java
bca3@chinarts-Veriton-M200-H410:~/Desktop$ java AccountClient Jimin

OUTPUT:

```
MENU
1.Deposit
2.Withdraw
3.Display
4.Exit
Enter your choice:1
Enter the amount to be deposited:
5000
Deposited Successfully
New Balance:5000.0

MENU
1.Deposit
2.Withdraw
3.Display
4.Exit
Enter your choice:2
Enter the amount to be withdrawn:
2500
Withdrawn Successfully
New Balance: 2500.0

MENU
1.Deposit
2.Withdraw
3.Display
4.Exit
Enter your choice:3
Name:Jimin
Current Balance:2500.0

MENU
1.Deposit
2.Withdraw
3.Display
4.Exit
Enter your choice:2
Enter the amount to be withdrawn:
2000
No Sufficient Balance

MENU
1.Deposit
2.Withdraw
3.Display
4.Exit
Enter your choice:3
Name:Jimin
```

```
Current Balance:2500.0

MENU
1.Deposit
2.Withdraw
3.Display
4.Exit
Enter your choice:4
```

7. SERVLET PROGRAM TO DISPLAY STUDENT DETAILS

AIM: Create an HTML form to read student details such as roll , name , age , sex , qualification , percentage of marks etc. Write a servlet program that displays the same details

PROGRAM:

File:Student.html

```
<html>
<head> <title> Mark list </title></head>
<body>
<form method=get action="http://localhost:8080/examples/StudentServlet">
Enter your roll.no:
<input type=text name=rollno size=20> <br><br>
Enter your name:
<input type=text name=name size=20> <br><br>
Enter your age:
<input type=text name=age size=20 ><br><br>
Enter your gender:
<input type=radio name=gender value=Female>Female
<input type=radio name=gender value=Male>Male<br><br>
Enter your qualification:
<input type=text name=qualification size=20> <br><br>
Enter the percentage:
<input type=text name=percentage size=20 ><br><br>
<input type=submit  value=Submit> <br>
</form>
</body>
</html>
```

File:StudentServlet.java

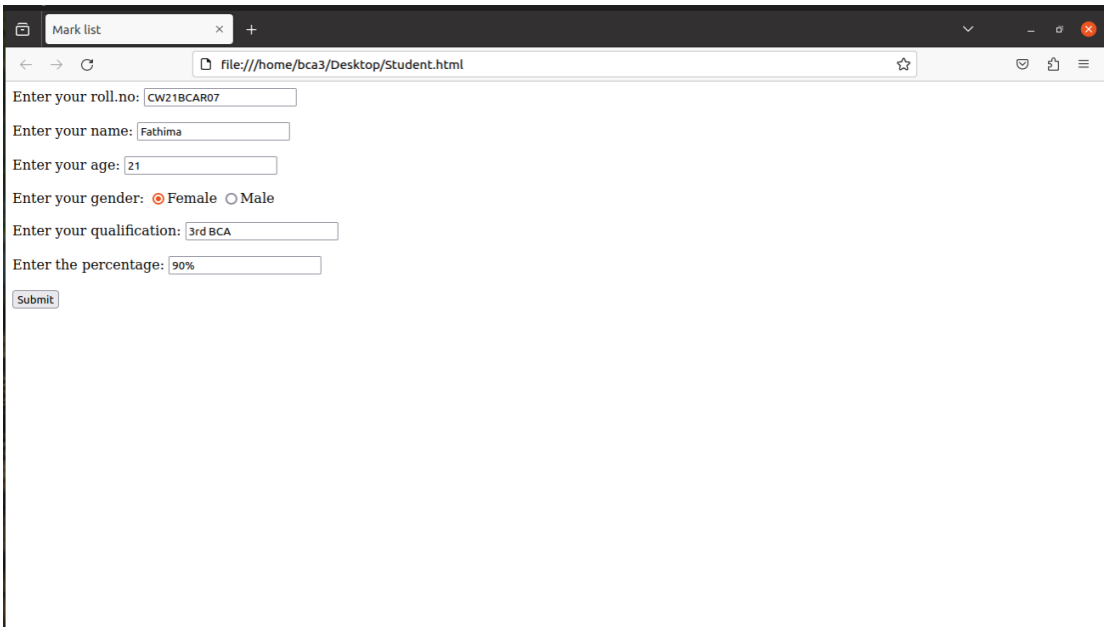
```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class StudentServlet extends HttpServlet
{
      public void doGet(HttpServletRequest req,HttpServletResponse resp)throws
IOException,ServletException
      {
            resp.setContentType("text/html");
            PrintWriter out=resp.getWriter();
            out.println("<html>");
            out.println("<head><title>Mark list</title></head>");

out.println("<body><h2>Roll.No:"+req.getParameter("rollno")+"<br>Name:"+req.getParameter("na
me")+"<br>Age:"+req.getParameter("age"));
```

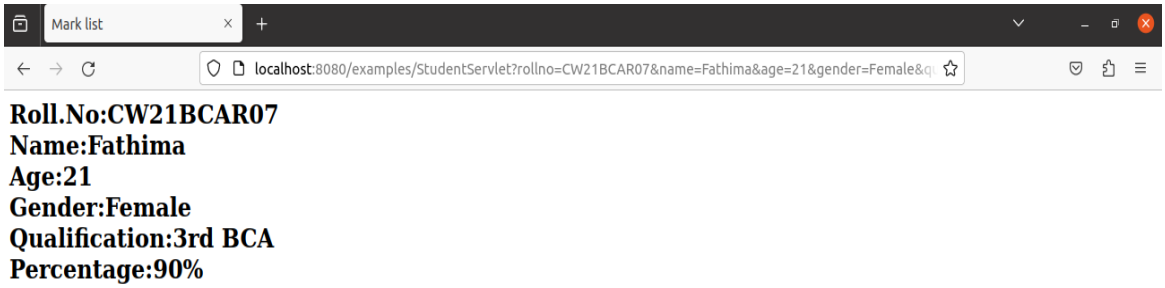
```
out.println("<br>Gender:"+req.getParameter("gender")+"<br>Qualification:"+req.getParameter("qua
lification")+"<br>Percentage:"+req.getParameter("percentage"));
            out.println("</h2></body></html>");
      }
}
```

Compile and Execute:

```
bca3@chinarts-desktop:~/Desktop/random$ vi StudentServlet.java
bca3@chinarts-desktop:~/Desktop/random$ locate servlet-api.jar
/usr/share/java/tomcat9-servlet-api.jar
/usr/share/tomcat9/lib/servlet-api.jar
bca3@chinarts-desktop:~/Desktop/random$ export
CLASSPATH=$CLASSPATH:/usr/share/java/tomcat9-servlet-api.jar
bca3@chinarts-desktop:~/Desktop/random$ javac StudentServlet.java
bca3@chinarts-desktop:~/Desktop/random$ chmod 777 StudentServlet.class
bca3@chinarts-desktop:~/Desktop/random$ cp StudentServlet.class /usr/share/tomcat9-
examples/examples/WEB-INF/classes/
bca3@chinarts-desktop:~/Desktop/random$ vi /usr/share/tomcat9-examples/examples/WEB-
INF/web.xml
bca3@chinarts-desktop:~/Desktop/random$ touch /usr/share/tomcat9-examples/examples/WEB-
INF/web.xml
```

OUTPUT:


Mark list
file:///home/bca3/Desktop/Student.html
Enter your roll.no: CW21BCAR07
Enter your name: Fathima
Enter your age: 21
Enter your gender: Female Male
Enter your qualification: 3rd BCA
Enter the percentage: 90%
Submit


Mark list
localhost:8080/examples/StudentServlet?rollno=CW21BCAR07&name=Fathima&age=21&gender=Female&q
Roll.No:CW21BCAR07
Name:Fathima
Age:21
Gender:Female
Qualification:3rd BCA
Percentage:90%

8. SERVLET PROGRAM TO DISPLAY THE CONTENTS OF A FILE

AIM: Write a servlet to display the contents of a file.

PROGRAM:

File:File.html

```
<html>
<head>
<title>File Servlet</title>
</head>
<body>
<form method=post action="http://localhost:8080/examples/FileServlet">
Enter the file name:
<input type=text name=filename size=20>
<input type=submit value="submit">
</form>
</body>
</html>
```

File:FileServlet.java

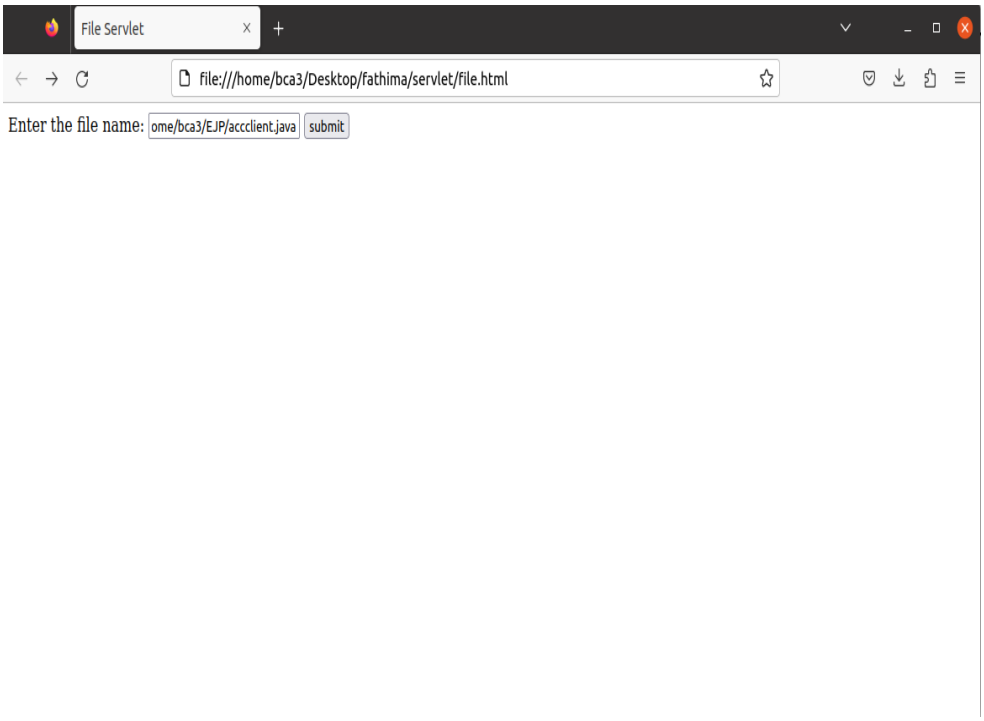
```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class FileServlet extends HttpServlet
{
    public void doPost(HttpServletRequest req,HttpServletResponse resp)throws
ServletException,IOException
    {
        File r;
        FileReader fr;
        BufferedReader br;
        try
        {
            r=new File(req.getParameter("filename"));
            fr=new FileReader(r);
            br=new BufferedReader(fr);
            if(!r.isFile())
            {
                resp.sendError(resp.SC_NOT_FOUND);
                return;
            }
        }
        catch(FileNotFoundException e)
        {
            resp.sendError(resp.SC_NOT_FOUND);
            return;
```

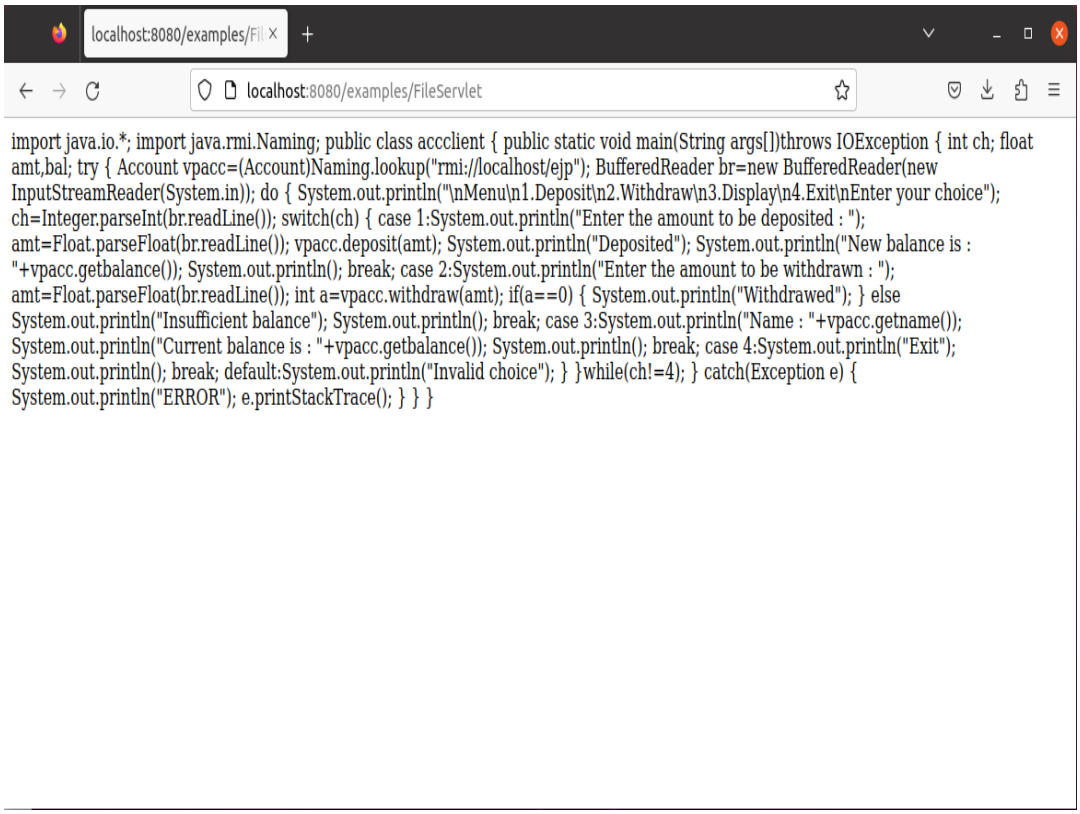
```
            }
            catch(SecurityException se)
            {
                  throw(new UnavailableException(this,"Servlet lacks appropriate priviledge"));
            }
            resp.setContentType("text/html");
            PrintWriter out=resp.getWriter();
            String  text;
            while((text=br.readLine())!=null)
                  out.println(text);
             br.close();
      }
}
```

Compile and Execute:

```
bca3@chinarts-Veriton-M200-H410:~/Desktop/servlet$ javac FileServlet.java -Xlint
bca3@chinarts-Veriton-M200-H410:~/Desktop/servlet$ chmod 777 FileServlet.class
bca3@chinarts-Veriton-M200-H410:~/Desktop/servlet$ cp FileServlet.class /usr/share/tomcat9-examples/examples/WEB-INF/classes/
bca3@chinarts-Veriton-M200-H410:~/Desktop/servlet$ vi /usr/share/tomcat9-examples/examples/WEB-INF/web.xml
bca3@chinarts-Veriton-M200-H410:~/Desktop/servlet$ touch /usr/share/tomcat9-examples/examples/WEB-INF/web.xml
```

OUTPUT:


File Servlet
file:///home/bca3/Desktop/fathima/servlet/file.html
Enter the file name: ome/bca3/EJP/accclient.java
submit


localhost:8080/examples/Fi
localhost:8080/examples/FileServlet
import java.io.*; import java.rmi.Naming; public class accclient { public static void main(String args[])throws IOException { int ch; float amt,bal; try { Account vpacc=(Account)Naming.lookup("rmi://localhost/ejp"); BufferedReader br=new BufferedReader(new InputStreamReader(System.in)); do { System.out.println("\nMenu\n1.Deposit\n2.Withdraw\n3.Display\n4.Exit\nEnter your choice"); ch=Integer.parseInt(br.readLine()); switch(ch) { case 1:System.out.println("Enter the amount to be deposited : "); amt=Float.parseFloat(br.readLine()); vpacc.deposit(amt); System.out.println("Deposited"); System.out.println("New balance is : "+vpacc.getbalance()); System.out.println(); break; case 2:System.out.println("Enter the amount to be withdrawn : "); amt=Float.parseFloat(br.readLine()); int a=vpacc.withdraw(amt); if(a==0) { System.out.println("Withdrawed"); } else System.out.println("Insufficient balance"); System.out.println(); break; case 3:System.out.println("Name : "+vpacc.getname()); System.out.println("Current balance is : "+vpacc.getbalance()); System.out.println(); break; case 4:System.out.println("Exit"); System.out.println(); break; default:System.out.println("Invalid choice"); } }while(ch!=4); } catch(Exception e) { System.out.println("ERROR"); e.printStackTrace(); } } }

9. SESSION HANDLING

AIM: Write a session handling servlet that displays total number of visits to that page.

PROGRAM:

File:VisitCounterServlet.java

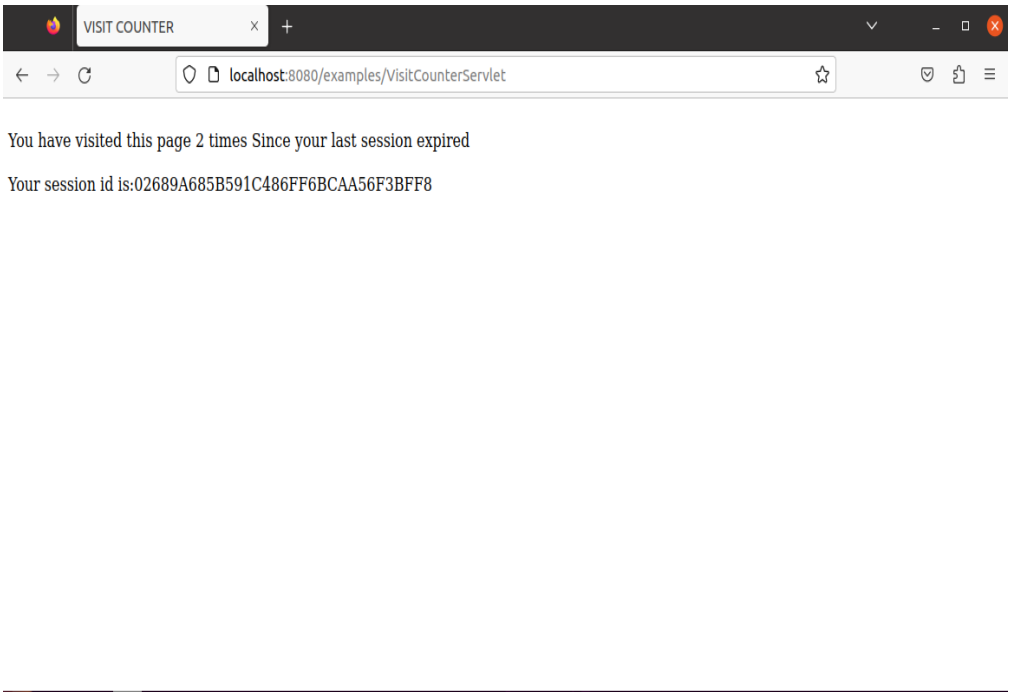
```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class VisitCounterServlet extends HttpServlet
{
	public void doGet(HttpServletRequest req,HttpServletResponse resp)throws
ServletException,IOException
	{
		resp.setContentType("text/html");
		PrintWriter out=resp.getWriter();
		HttpSession thisUser=req.getSession(true);
		Integer visits;
		if(!thisUser.isNew())
		{
			visits=(Integer)thisUser.getValue("visitcounter.visits");
			if(visits==null)
				visits=new Integer(1);
			else
				visits=new Integer(visits.intValue()+1);
		}
		else
			visits=new Integer(1);
		thisUser.putValue("visitcounter.visits",visits);
		out.println("<html><head><title>VISIT COUNTER</title></head>");
		out.println("<body><br>You have visited this page "+visits+" times");
		out.println("Since your last session expired");
		out.println("<br><br>Your session id is:"+thisUser.getId()+"</body></html>");
	}
}
```

Compile and Execute:

```
bca3@chinarts-Veriton-M200-H410:~/Desktop/servlet$ locate servlet-api.jar
/usr/share/java/tomcat9-servlet-api.jar
/usr/share/tomcat9/lib/servlet-api.jar
bca3@chinarts-Veriton-M200-H410:~/Desktop/servlet$ export
CLASSPATH=$CLASSPATH:/usr/share/tomcat9/lib/servlet-api.jar
bca3@chinarts-Veriton-M200-H410:~/Desktop/servlet$ javac VisitCounterServlet.java -Xlint
bca3@chinarts-Veriton-M200-H410:~/Desktop/servlet$ cp VisitCounterServlet.class /usr/share/tomcat9-
examples/examples/WEB-INF/classes/
```

bca3@chinarts-Veriton-M200-H410:~/Desktop/servlet$ vi /usr/share/tomcat9-examples/examples/WEB-INF/web.xml
bca3@chinarts-Veriton-M200-H410:~/Desktop/servlet$ touch /usr/share/tomcat9-examples/examples/WEB-INF/web.xml

OUTPUT:


VISIT COUNTER
localhost:8080/examples/VisitCounterServlet
You have visited this page 2 times Since your last session expired
Your session id is:02689A685B591C486FF6BCAA56F3BFF8

10. CORBA ARITHMETIC OPERATIONS

AIM: Write a corba program to perform arithmetic operations.

PROGRAM:

File:Arith.idl

```
interface Arith
{
float sum(in float a,in float b);
float sub(in float a,in float b);
float mul(in float a,in float b);
float div(in float a,in float b);
};
```

Compile:
$idlj -fall Arith.idl

File:ArithImpl.java

```
public class ArithImpl extends ArithPOA
{
	public ArithImpl(){ }
	public float sum(float a,float b)
	{
		float c=a+b;
		return c;
	}
	public float sub(float a,float b)
	{
		float c=a-b;
		return c;
	}
	public float mul(float a,float b)
	{
		float c=a*b;
		return c;
	}
	public float div(float a,float b)
	{
		float c=a/b;
		return c;
	}
}
```

Compile:
$javac ArithImpl.java

File: vi ArithInitPOA

```
import org.omg.CORBA.*;
import org.omg.CosNaming.*;
import org.omg.PortableServer.POA;
public class ArithInitPOA
{
	public static void main(String args[])
	{
		try
		{
			ORB myOrb=ORB.init(args,null);
			ArithImpl ob=new ArithImpl();
			POA rootPOA=(POA)myOrb.resolve_initial_references("RootPOA");
			rootPOA.the_POAManager().activate();
			org.omg.CORBA.Object obj=rootPOA.servant_to_reference(ob);
			Arith acct=(Arith)ArithHelper.narrow(obj);
			org.omg.CORBA.Object objref=myOrb.resolve_initial_references("NameService");
			NamingContextExt nc=NamingContextExtHelper.narrow(objref);
			NameComponent[] name=nc.to_name(args[0]);
			nc.rebind(name,acct);
			System.out.println("Object registered under the name "+args[0]);
			myOrb.run();
		}
		catch(Exception e)
		{
			System.out.println("ERROR");
			e.printStackTrace();
		}
	}
}
```

Compile and Execute:

```
$javac ArithInitPOA.java
$tnameserv &
$java ArithInitPOA  JIMIN
Object registered under the name JIMIN
```

File:ArithClient.java

```
import org.omg.CORBA.*;
import org.omg.CORBA.ORBPackage.*;
import org.omg.CosNaming.*;
import java.util.*;
import java.io.*;
public class ArithClient
{
	public static void main (String args[])throws IOException
	{
```

```
        ORB orb=ORB.init(args,null);
        org.omg.CORBA.Object ref=null;
        Arith obj;
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        try
        {
            ref=orb.resolve_initial_references("NameService");
        }
        catch(InvalidName e)
        {
            System.out.println("Could not locate name");
            System.exit(0);
        }
        NamingContext nc=NamingContextHelper.narrow(ref);
        NameComponent comp=new NameComponent(args[0],"");
        NameComponent[] path={comp};
        try
        {
            ref=nc.resolve(path);
            obj=ArithHelper.narrow(ref);
            int ch;
            float n1,n2,res;
            System.out.println("Enter two real numbers");
            n1=Float.parseFloat(br.readLine());
            n2=Float.parseFloat(br.readLine());
            do
            {

System.out.print("\nMENU\n1.Addition\n2.Subtraction\n3.Multiplication\n4.Division\n5.Exit\nEnte
r your choice: ");
                ch=Integer.parseInt(br.readLine());
                switch(ch)
                {
                    case 1: res=obj.sum(n1,n2);
                        System.out.println("Sum: "+res);
                        break;
                    case 2: res=obj.sub(n1,n2);
                        System.out.println("Difference: "+res);
                        break;
                    case 3: res=obj.mul(n1,n2);
                        System.out.println("Product: "+res);
                        break;
                    case 4: res=obj.div(n1,n2);
                        System.out.println("Quotient: "+res);
                        break;
                    case 5: System.exit(0);
                    default: System.out.println("Enter a valid choice\n");
                }
            }while(ch!=5);
        }
```

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

Compile and Execute:

$javac ArithClient.java
$java ArithClient JIMIN

OUTPUT:

```
Enter two real numbers
7
3

MENU
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Exit
Enter your choice: 1
Sum: 10.0

MENU
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Exit
Enter your choice: 2
Difference: 4.0

MENU
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Exit
Enter your choice: 3
Product: 21.0

MENU
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Exit
Enter your choice: 4
Quotient: 2.3333333

MENU
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Exit
Enter your choice: 5
```