ENTERPRISE JAVA PROGRAMMING

GOVT. COLLEGE THALASSERY

KANNUR UNIVERSITY

CERTIFICATE

This is to certify that this is the bonafide record of the original work done by Ms/Mr. …….. Reg no.....…….... of Sixth semester BCA in the programming in ENTERPRISE JAVA PROGRAMMING lab during the year 2022-23.

Head of the Department Lecture in Charge

Submitted for the university Practical Examination held on…….

Examiner 1:

Examiner 2:

PROGRAM 1

AIM: JDBC program to insert, Delete and Update records into Employee table.

```
import java.sql.*;
import java.io.*;
class JDBCPGM1
{
    public static void main(String arg[])
    {
        ResultSet rs;
        ResultSetMetaData rm;
        String eno,ename,salary;
        int ch,n;
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/emp","root","Gct@2018");
            Statement stmt=con.createStatement();
            InputStreamReader isr=new InputStreamReader(System.in);
            BufferedReader br=new BufferedReader(isr);
            System.out.println("1) SELECT\n2) INSERT\n3) UPDATE \n4) DELETE\n5) EXIT");
            while(true)
            {
                System.out.println("\nEnter your choice:");
                ch=Integer.parseInt(br.readLine());
                System.out.print("************************ \n");
                switch(ch)
                {
                    case 1:rs=stmt.executeQuery("select * from employee");
                        rm=rs.getMetaData();
                        n=rm.getColumnCount();
                        while(rs.next())
                        {
                            for(int i=1;i<=n;i++)
                            {
                                System.out.print(rs.getString(i)+" \t");
                            }
                             System.out.println();
                        }
                        break;

                    case 2:System.out.println("Enter employee number :");
                        eno=br.readLine();
                        System.out.println("Enter Name :");
                        ename=br.readLine();
                        System.out.println("Enter salary :");
                        salary=br.readLine();
                        stmt.execute("insert into employee values ("+eno+",'"+ename+"',"+salary+")");
                        System.out.println("One record inserted!");
                        break;
```

```
                    case 3:System.out.println("Enter employee number to edit :");
                            eno=br.readLine();
                            System.out.println("Enter Name :");
                            ename=br.readLine();
                            System.out.println("Enter salary :");
                            salary=br.readLine();
                        stmt.execute("update employee set ename='"+ename+"',salary="+salary+" where eno="+eno+"");
                            System.out.println("One record Updated!");
                            break;


                    case 4:System.out.println("Enter employee number to delete :");
                            eno=br.readLine();
                            stmt.execute("delete from employee where eno="+eno+"");
                            System.out.println("One record Deleted!");
                            break;

                    case 5:br.close();
                            con.close();
                            System.exit(0);
                    }
                    System.out.print("************************ \n");
                }
            }
            catch(Exception e)
            {
                System.out.println(e);
            }
    }
}
```

OUTPUT :

```
student@gct12-Veriton-M200-H110:~/Desktop/EJP record/PGM1$ javac JDBCPGM1.java
student@gct12-Veriton-M200-H110:~/Desktop/EJP record/PGM1$ java JDBCPGM1
Tue Jan 03 13:49:14 YAKT 2023 WARN: Establishing SSL connection without server's identity verification is not recommended. According to MySQL
5.5.45+, 5.6.26+ and 5.7.6+ requirements SSL connection must be established by default if explicit option isn't set. For compliance with exist
ing applications not using SSL the verifyServerCertificate property is set to 'false'. You need either to explicitly disable SSL by setting us
eSSL=false, or set useSSL=true and provide truststore for server certificate verification.
1) SELECT
2) INSERT
3) UPDATE
4) DELETE
5) EXIT

Enter your choice:
1
************************
1       Shijin  30000
2       Sarang  10000
3       Paul    40000
4       jishnu  890000
5       maya    897000
************************

Enter your choice:
2
************************
Enter employee number :
6
Enter Name :
Rahul
Enter salary :
50000
One record inserted!
************************
```

```
Enter your choice:
3
************************
Enter employee number to edit :
5
Enter Name :
John
Enter salary :
60000
One record Updated!
************************

Enter your choice:
4
************************
Enter employee number to delete :
4
One record Deleted!
************************

Enter your choice:
1
************************
1       Shijin  30000
2       Sarang  10000
3       Paul    40000
5       John    60000
6       Rahul   50000
************************

Enter your choice:
5
************************
student@gct12-Veriton-M200-H110:~/Desktop/EJP record/PGM1$
```

PROGRAM 2

AIM: JDBC program to connect to Student table. Implement the record scrolling functions – first(), last(), next(), previous(), beforeFirst(), afterLast(), absolute() and relative().

```
import java.sql.*;
class JDBCPGM2
{
    public static void main(String args[])
    {
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
 Connection con=DriverManager.getConnection( "jdbc:mysql://localhost:3306/student123","root","Gct@2018");
   Statement stmt =con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_READ_ONLY);
            ResultSet rs=stmt.executeQuery("select * from student");
            System.out.println("\n-->Print records from bottom to top");
            rs.afterLast();
            while(rs.previous())
            {
                System.out.println(rs.getString(1)+"\t "+rs.getInt(2)+"\t "+rs.getString(3));
            }
            System.out.println("\n-->To print 3rd record data using absolute(3):");
            rs.absolute(3);
            System.out.println(rs.getString(1)+"\t "+rs.getInt(2)+"\t "+rs.getString(3));
            System.out.println("\n-->Print 2rd record data using relative(-1)");
            rs.relative(-1);
            System.out.println(rs.getString(1)+"\t "+rs.getInt(2)+"\t "+rs.getString(3));
            System.out.println("\n-->Print the first record after moving to first position with first():");
            rs.first();
            System.out.println(rs.getString(1)+"\t "+rs.getInt(2)+"\t "+rs.getString(3));
            System.out.println("\n-->Print the last record after moving to last record using last():");
            rs.last();
            System.out.println(rs.getString(1)+"\t "+rs.getInt(2)+"\t "+rs.getString(3));
            System.out.println("\n-->Print records from top to bottom");
            rs.beforeFirst();
            while(rs.next())
            {
                System.out.println(rs.getString(1)+"\t "+rs.getInt(2)+"\t "+rs.getString(3));
            }
            System.out.println("\n");
            rs.close();
            stmt.close();
            con.close();
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

OUTPUT:

```
student@gct12-Veriton-M200-H110:~/Desktop/EJP record/PGM2$ javac JDBCPGM2.java
student@gct12-Veriton-M200-H110:~/Desktop/EJP record/PGM2$ java JDBCPGM2
Tue Dec 06 14:32:08 YAKT 2022 WARN: Establishing SSL connection without server's identity verification is not recommended. According to MySQL
5.5.45+, 5.6.26+ and 5.7.6+ requirements SSL connection must be established by default if explicit option isn't set. For compliance with exist
ing applications not using SSL the verifyServerCertificate property is set to 'false'. You need either to explicitly disable SSL by setting us
eSSL=false, or set useSSL=true and provide truststore for server certificate verification.

-->Print records from bottom to top
ramya    375     female
ramya    375     female
ramanan  375     female
ramu     345     male
ram      23      male

-->To print 3rd record data using absolute(3):
ramanan  375     female

-->Print 2rd record data using relative(-1)
ramu     345     male

-->Print the first record after moving to first position with first():
ram      23      male

-->Print the last record after moving to last record using last():
ramya    375     female

-->Print records from top to bottom
ram      23      male
ramu     345     male
ramanan  375     female
ramya    375     female
ramya    375     female
```

PROGRAM 3

AIM: JDBC program to display database metadata.

```
import java.sql.*;
class JDBCPGM4
{
    public static void main(String arg[])
    {
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/gct","root","Gct@2018");
            DatabaseMetaData dbmd=con.getMetaData();
            System.out.println("Driver Name : "+ dbmd.getDriverName());
            System.out.println("Driver Version : "+ dbmd.getDriverVersion());
            System.out.println("User Name : "+ dbmd.getUserName());
            System.out.println("Database Product Name : "+ dbmd.getDatabaseProductName());
            System.out.println("Database Product Version : "+ dbmd.getDatabaseProductVersion());
            con.close();
        }
        catch(Exception e)
        {
              System.out.println(e);
         }
    }
}
```

OUTPUT:

```
student@gct12-Veriton-M200-H110:~/Desktop/EJP record$ javac JDBCPGM4.java
student@gct12-Veriton-M200-H110:~/Desktop/EJP record$ java JDBCPGM4
Mon Dec 05 15:01:11 YAKT 2022 WARN: Establishing SSL connection without server's identity verification is not recommended. According to MySQL
5.5.45+, 5.6.26+ and 5.7.6+ requirements SSL connection must be established by default if explicit option isn't set. For compliance with exist
ing applications not using SSL the verifyServerCertificate property is set to 'false'. You need either to explicitly disable SSL by setting us
eSSL=false, or set useSSL=true and provide truststore for server certificate verification.
Driver Name : MySQL Connector Java
Driver Version : mysql-connector-java-5.1.42 ( Revision: ${revinfo.commit} )
User Name : root@localhost
Database Product Name : MySQL
Database Product Version : 5.7.22-0ubuntu0.17.10.1
student@gct12-Veriton-M200-H110:~/Desktop/EJP record$
```

PROGRAM 4

AIM: JDBC program to display Resultset metadata.

```
import java.sql.*;
class JDBCPGM5
{
    public static void main(String arg[])
    {
       try
       {
           Class.forName("com.mysql.jdbc.Driver");
           Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/gct","root","Gct@2018");
           PreparedStatement ps=con.prepareStatement("Select * From BCA");
           ResultSet rs=ps.executeQuery();
           ResultSetMetaData rsmd=rs.getMetaData();
           System.out.println("Total Columns : "+ rsmd.getColumnCount());
           System.out.println("Column Name of First Column : "+ rsmd.getColumnName(1));
           System.out.println("Column Type Name of First Column : "+ rsmd.getColumnTypeName(1));
           rs.close();
           ps.close();
           con.close();
       }
       catch(Exception e)
       {
             System.out.println(e);
        }
    }
}
```

OUTPUT:

PROGRAM 5

Aim: RMI program for Complex number operation.

Complex.java

```
import java.rmi.*;
import java.rmi.server.*;
import java.io.*;
public class Complex implements Serializable
{
	int real,imag;
	public Complex(int a,int b)
	{
		real=a;
		imag=b;
	}
}
```

Complexi.java

```
import java.rmi.*;
import java.rmi.server.*;
public interface Complexi extends Remote
{
	public Complex add(Complex c1,Complex c2)throws Exception;
	public Complex substract(Complex c1,Complex c2)throws Exception;
	public Complex multiply(Complex c1,Complex c2)throws Exception;
}
```

Complexc.java

```
import java.rmi.*;
import java.rmi.server.*;
import java.io.*;
public class Complexc extends UnicastRemoteObject implements Complexi
{
	Complex cs;

	public Complexc()throws RemoteException,IOException
	{
		cs=new Complex(0,0);
	}

	public Complex add(Complex c1,Complex c2)throws Exception
	{
		cs.real=c1.real+c2.real;
		cs.imag=c1.imag+c2.imag;
		return cs;
	}
```

```
    public Complex substract(Complex c1,Complex c2)throws Exception
    {
        cs.real=c1.real-c2.real;
        cs.imag=c1.imag-c2.imag;
        return cs;
    }
    public Complex multiply(Complex c1,Complex c2)throws Exception
    {
        cs.real=c1.real*c2.real-c1.imag*c2.imag;
        cs.imag=c1.real*c2.imag+c1.imag*c2.real;
        return cs;
    }
}
```

Server.java

```
import java.rmi.*;
import java.rmi.server.*;
import java.io.*;
public class Server
{
    public static void main(String args[])
    {
        try
        {
            Complexc cs=new Complexc();
            Naming.rebind("rmi",cs);
            System.out.println("server is ready!");
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

6

```
import java.rmi.*;
import java.io.*;
public class Client
{
    public static void main(String args[])
    {
        try
        {
            Complexi com=(Complexi)Naming.lookup("rmi");
            InputStreamReader isr=new InputStreamReader(System.in);
            BufferedReader br=new BufferedReader(isr);
            System.out.println("\nEnter real and imaginary part of first number");
            int r1=Integer.parseInt(br.readLine());
            int i1=Integer.parseInt(br.readLine());
            System.out.println("\nEnter real and imaginary part of second number");
```

```
                int r2=Integer.parseInt(br.readLine());
                int i2=Integer.parseInt(br.readLine());
                Complex cs1=new Complex(r1,i1);
                Complex cs2=new Complex(r2,i2);
                Complex cs3=new Complex(0,0);
                System.out.println("\nResult:");
                cs3=com.add(cs1,cs2);
                System.out.println("\nSum="+cs3.real+"+"+cs3.imag+"i");
                cs3=com.substract(cs1,cs2);
                System.out.println("\nDifference="+cs3.real+"+"+cs3.imag+"i");
                cs3=com.multiply(cs1,cs2);
                System.out.println("\nproduct="+cs3.real+"+"+cs3.imag+"i");
        }
        catch(Exception e)
        {
                System.out.println("\nException="+e);
        }
}
}
```

OUTPUT:

```
student@gct12-Veriton-M200-H110:~/Desktop/EJP record/PGM6$ javac *.java
student@gct12-Veriton-M200-H110:~/Desktop/EJP record/PGM6$ rmic Complexc
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.
student@gct12-Veriton-M200-H110:~/Desktop/EJP record/PGM6$ rmiregistry&
[1] 4057
student@gct12-Veriton-M200-H110:~/Desktop/EJP record/PGM6$ java Server
server is ready!
```

```
student@gct12-Veriton-M200-H110:~/Desktop/EJP record/PGM6$ java Client

Enter real and imaginary part of first number
10
20

Enter real and imaginary part of second number
10
20

Result:

Sum=20+40i

Difference=0+0i

product=-300+400i
student@gct12-Veriton-M200-H110:~/Desktop/EJP record/PGM6$
```

PROGRAM 6

AIM: RMI program for Bank operation.

BankC.java

```
import java.rmi.*;
import java.rmi.server.*;
public class Bankc extends UnicastRemoteObject implements Banki
{
	String name;
	float bal;
	public Bankc() throws RemoteException
	{
		super();
		name="John";
		bal=0;
	}
	public String getName() throws RemoteException
	{
		return name;
	}
	public float balance()throws RemoteException
	{
		return bal;
	}
	public void withdraw(float amt)throws RemoteException
	{
		bal=bal-amt;
	}
	public void deposit(float amt)throws RemoteException
	{
		bal=bal+amt;
	}
}
```

BankI.java

```
import java.rmi.*;
public interface Banki extends Remote
{
	String getName()throws RemoteException;
	float balance()throws RemoteException;
	void withdraw(float amt)throws RemoteException;
	void deposit(float amt)throws RemoteException;
}
```

Client.java

```
import java.rmi.*;
import java.rmi.server.*;
import java.rmi.registry.*;
import java.io.*;
public class Client
{
	public static void main(String args[])
	{
	    try
	    {
		Banki stub=(Banki)Naming.lookup("rmi://127.0.0.1:1099/bank");
		while(true)
		{
		    InputStreamReader isr=new InputStreamReader(System.in);
		    BufferedReader br=new BufferedReader(isr);
		    System.out.println("\n1.Account holder\n2.Deposit\n3.Withdraw\n4.balance\n5.Exit\n");
		    System.out.print("Enter your choice:");
		    int i=Integer.parseInt(br.readLine());
		    switch(i)
		    {
			case 1:String s=stub.getName();
			    System.out.println("Account Holder is "+s);
			     break;

			case 2:System.out.println("Enter the amount to be deposited");
				float a=Float.parseFloat(br.readLine());
				stub.deposit(a);
				float db=stub.balance();
				System.out.println("Balance="+db);
				break;

			case 3:System.out.println("Enter the amount to be withdrawed");
				 float w=Float.parseFloat(br.readLine());
				 float wb=stub.balance();
				  if(wb<w)
				       System.out.println("Insuffient balance\n");
				 else
				      stub.withdraw(w);
				  float wb1=stub.balance();
				  System.out.println("Balance="+wb1);
				  break;

			case 4:float b=stub.balance();
				  System.out.println("Balance="+b);
				  break;

			       case 5:System.exit(0);
			}
		}
	    }
	catch(Exception e)
	{
```

```
                System.out.println(e);
        }
        }
}
```

Server.java

```
import java.rmi.*;
import java.rmi.server.*;
import java.rmi.registry.*;
public class Server
{
        public static void main(String args[])
        {
        try{
                Bankc ob=new Bankc();
                Naming.rebind("rmi://127.0.0.1:1099/bank",ob);
                System.out.println("server is ready");
           }
        catch(Exception e)
        {
                System.out.println(e);
        }
        }
}
```

OUTPUT:

```
student@gct12-Veriton-M200-H110:~/Desktop/EJP record/PGM7$ javac *.java
student@gct12-Veriton-M200-H110:~/Desktop/EJP record/PGM7$ rmic Bankc
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.
student@gct12-Veriton-M200-H110:~/Desktop/EJP record/PGM7$ rmiregistry&
[1] 3200
student@gct12-Veriton-M200-H110:~/Desktop/EJP record/PGM7$ java Server
server is ready
```

```
student@gct12-Veriton-M200-H110:~/Desktop/EJP record/PGM7$ java Client

1.Account holder
2.Deposit
3.Withdraw
4.balance
5.Exit

Enter your choice:1
Account Holder is John

1.Account holder
2.Deposit
3.Withdraw
4.balance
5.Exit

Enter your choice:2
Enter the amount to be deposited
10000
Balance=19500.0

1.Account holder
2.Deposit
3.Withdraw
4.balance
5.Exit

Enter your choice:4
Balance=19500.0

1.Account holder
2.Deposit
3.Withdraw
4.balance
5.Exit
```

```
Enter your choice:4
Balance=19500.0

1.Account holder
2.Deposit
3.Withdraw
4.balance
5.Exit

Enter your choice:3
Enter the amount to be withdrawed
1000
Balance=18500.0

1.Account holder
2.Deposit
3.Withdraw
4.balance
5.Exit

Enter your choice:4
Balance=18500.0

1.Account holder
2.Deposit
3.Withdraw
4.balance
5.Exit

Enter your choice:5
student@gct12-Veriton-M200-H110:~/Desktop/EJP record/PGM7$
```

PROGRAM 7

AIM: Create an HTML form that reads a file name from the user. Write a servlet program that displays the contents of the file, specified by the user.

FileServlet.html

```
<html>
    <body>
        <form method=get action=”FileServlet”>
            User name:<input type=”text” name=”user”>
            <input type=”file” name=”filename”>
            <input type=”submit” value=”login”>
        </form>
    </body>
</html>
```

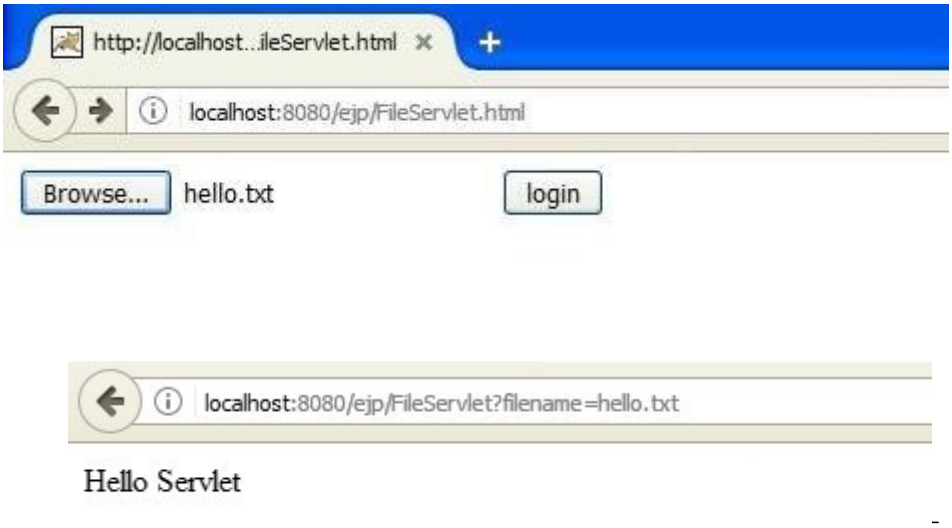
FileServlet.java

```
import javax.servlet.ServletException;
import javax.servlet.http.*;
import java.io.*;
public class FileServlet extends HttpServlet
{
    public void service(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException
    {
        res.setContentType("text/html");
        PrintWriter pw = res.getWriter();
        String name = req.getParameter("filename");
        BufferedReader br = new BufferedReader(new FileReader("c:/"+name));
        String str;
        while( (str = br.readLine()) != null )
        {
            pw.println(str + "<BR>");
        }
        br.close();
        pw.close();
    }
}
```

web.xml

```
<servlet>
    <servlet-name>FileServlet</servlet-name>
    <servlet-class>FileServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>FileServlet</servlet-name>
    <url-pattern>/FileServlet</url-pattern>
</servlet-mapping>
```

OUTPUT:


http://localhost...ileServlet.html
localhost:8080/ejp/FileServlet.html
Browse...
hello.txt
login

localhost:8080/ejp/FileServlet?filename=hello.txt
Hello Servlet

PROGRAM 8

AIM: Create a html form to read student details such as Roll ,Name ,Age ,Sex ,Qualification ,Percentage of mark ,write a servlet program that display same details

Student.html

```
<html>
    <body>
        <form action="Student" method="GET">
                Name: <input type="text" name="name">
                <br />
                Roll No: <input type="text" name="roll" />
                <br />
                Age: <input type="text" name="age">
                <br />
                Sex: <input type="text" name="sex" />
                <br />
               Qualification:<input type="text" name="qualification">
               <br />
               Percetage of mark: <input type="text" name="percent" />
               <br />
               <input type="submit" value="Submit" />
        </form>
     </body>
</html>
```

Student.java

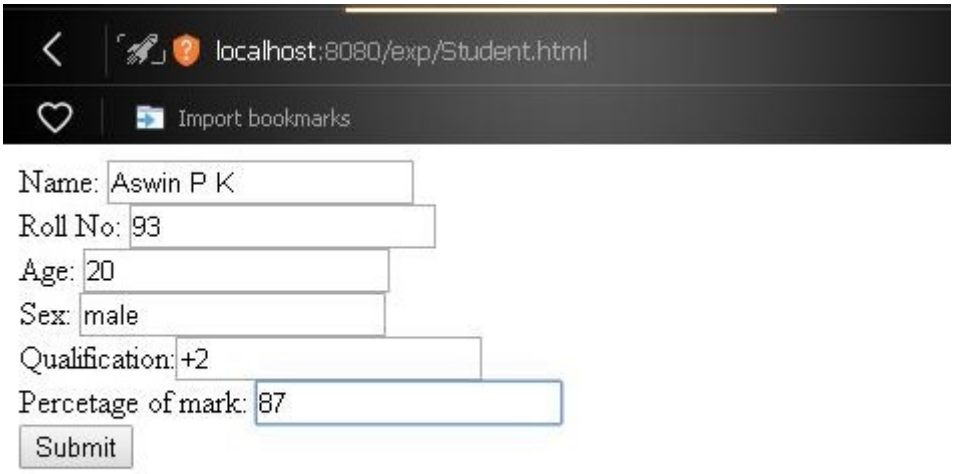
```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Student extends HttpServlet
{
      public void doGet(HttpServletRequest req,HttpServletResponse res) throws ServletException,IOException
      {
            res.setContentType("text/html");
            PrintWriter out = res.getWriter();
            String title = "Using GET Method to Read Form Data";
            out.println( "<html>");
            out.println("<b>Name</b>: ");
            out.println(request.getParameter("name") );
            out.println("<b>Roll No</b>: ");
            out.println(request.getParameter("roll"));
            out.println("<b>Age</b>: ");
            out.println(request.getParameter("age") );
            out.println("<b>Sex</b>: ");
            out.println(request.getParameter("sex") );
            out.println("<b>Qualification</b>: ");
```

```
            out.println(request.getParameter("qualification"));
            out.println("<b>Percetage of mark:</b>: ");
            out.println(request.getParameter("percent"));
            out.println("</body></html>");
        }
}
```

web.xml

```
<servlet>
        <servlet-name> Student</servlet-name>
        <servlet-class> Student </servlet-class>
</servlet>
<servlet-mapping>
        <servlet-name> Student </servlet-name>
        <url-pattern>/ Student </url-pattern>
</servlet-mapping>
```

OUTPUT:



Name: Aswin P K **Roll No**: 93 **Age**: 20 **Sex**: male **Qualification**: +2 **Percetage of mark:**: 87

PROGRAM 9

AIM: Session handling servlet that displays total number of visits to that page.

SessionTrack.java

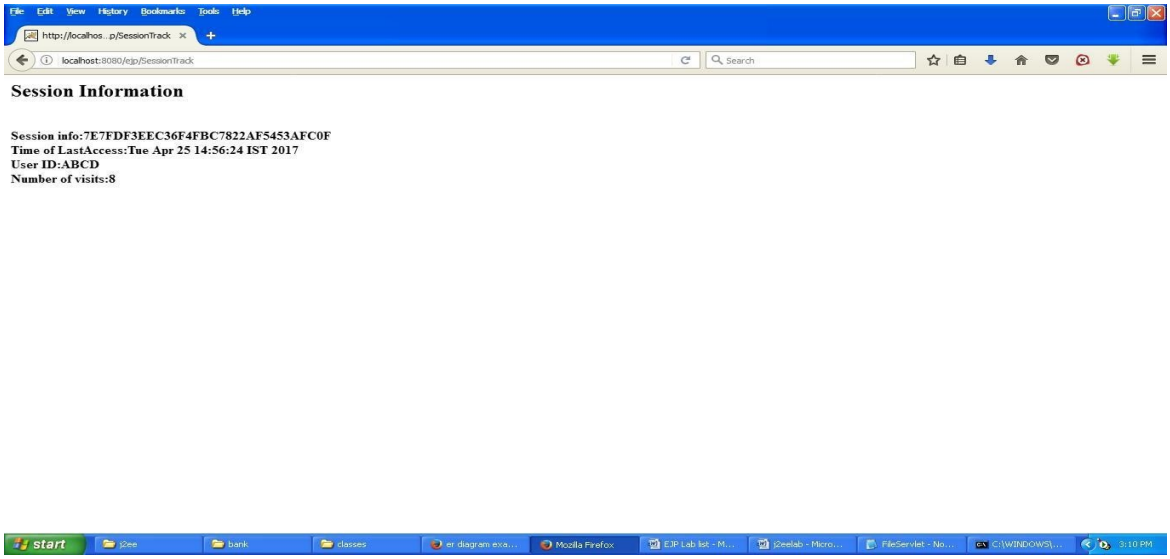
```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
public class SessionTrack extends HttpServlet
{
	public void doGet(HttpServletRequest req,HttpServletResponse res) throws ServletException,IOException
	{
		HttpSession session=request.getSession(true);
		Date creationTime=new Date(session.getCreationTime());
		Date lastAccessTime=new Date(session.getLastAccessedTime());
		String title="Welcome back to my website";
		Integer visitCount=new Integer(0);
		String visitCountKey=new String("visitCount");
		String userIDKey=new String("userID");
		String userID=new String("ABCD");
		if(session.isNew())
		{
			session.setAttribute(userIDKey,userID);
		}
		else
		{
			visitCount=(Integer)session.getAttribute(visitCountKey);
			visitCount=visitCount+1;
			userID=(String)session.getAttribute(userIDKey);
		}
		session.setAttribute(visitCountKey,visitCount);
		response.setContentType("text/html");
		PrintWriter out=response.getWriter();
		out.println("<h2><b>Session Information</h2>");
		out.println("<br><b>Session info:"+session.getId());
		out.println("<br><b>Time of LastAccess:"+lastAccessTime);
		out.println("<br><b>User ID:"+userID);
		out.println("<br><b>Number of visits:"+visitCount);
	}
}
```

web.xml

```
<servlet>
    <servlet-name> SessionTrack</servlet-name>
    <servlet-class> SessionTrack </servlet-class>
</servlet>
```

```
<servlet-mapping>
    <servlet-name> SessionTrack </servlet-name>
    <url-pattern>/ SessionTrack </url-pattern>
</servlet-mapping>
```

OUTPUT:



PROGRAM 10

AIM: CORBA Program for Arithmetic Operation.

Arithmetic.idl

```
interface Arithmetic
{
        float add(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);
};
```

ArithmeticImp.java

```
public class ArithmeticImp extends _ArithmeticImplBase
{
        float c;
        public float add(float a,float b)
        {
                c=a+b;
                return c;
        }
        public float sub(float a,float b)
        {
                c=a-b;
                return c;
}
public float mul(float a,float b)
{
        c=a*b;                  return c;
        }
        public float div(float a,float b)
        {
                c=a/b;
                return c;
        }
}
```

Client.java

```
import org.omg.CORBA.*;
import org.omg.CosNaming.*;
import java.io.*;
public class Client
{
        public static void main(String arg[])
        {
            try
            {
                float a,b,c;
                BufferedReader rd=new BufferedReader(new InputStreamReader(System.in));
                ORB orb=ORB.init(arg,null);
                org.omg.CORBA.Object ob=orb.resolve_initial_references("NameService");
                NamingContext ctx=NamingContextHelper.narrow(ob);
                NameComponent nc=new NameComponent("Message","");
                NameComponent path[]={nc};
                Arithmetic ar=ArithmeticHelper.narrow(ctx.resolve(path));
                System.out.println("Enter two numbers");
                a=Float.parseFloat(rd.readLine());
                b=Float.parseFloat(rd.readLine());
                c=ar.add(a,b);
                System.out.println("Sum="+c);
                c=ar.sub(a,b);
                System.out.println("Substract="+c);
                c=ar.mul(a,b);
                System.out.println("product="+c);
                c=ar.div(a,b);
                System.out.println("division="+c);
        }
        catch(Exception e)
        {
                System.out.println(e);
        }
}
```

Server.java:

```
import org.omg.CORBA.*;
import org.omg.CosNaming.*;
public class Server
{
	public static void main(String arg[])
	{
		try
		{
			ORB orb=ORB.init(arg,null);
			org.omg.CORBA.Object ob=orb.resolve_initial_references("NameService");
			NamingContext ctx=NamingContextHelper.narrow(ob);
			NameComponent nc=new NameComponent("Message","");
			NameComponent path[]={nc};
			ArithmeticImp m=new ArithmeticImp();
			ctx.rebind(path,m);
			orb.run();
		}
		catch(Exception e)
		{
			System.out.println(e);
		}
}
```

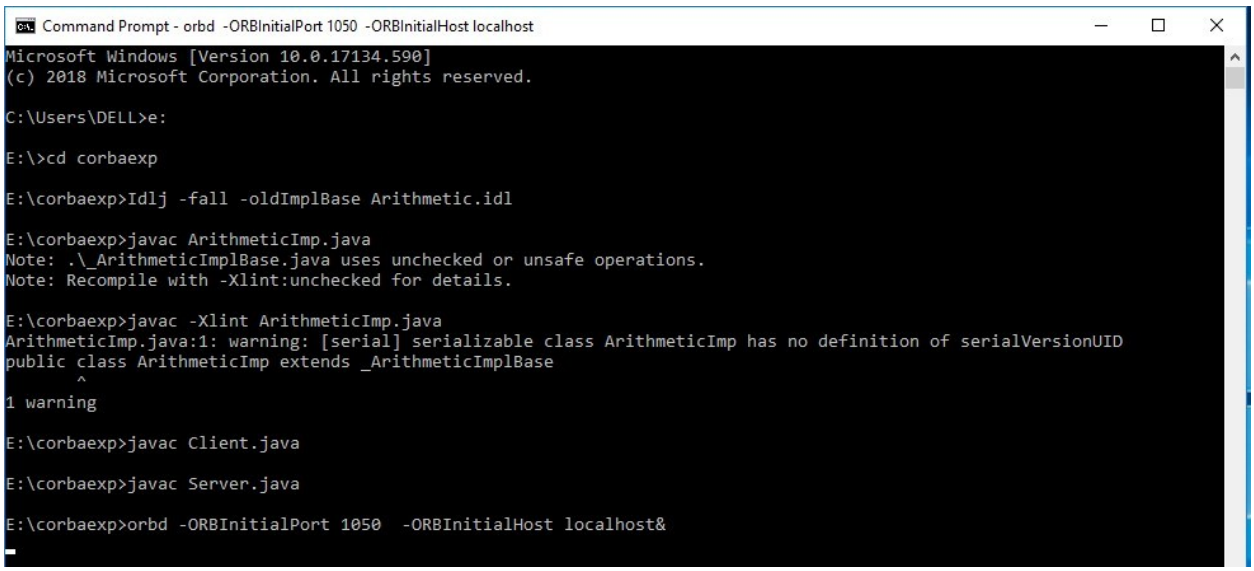
To compile:

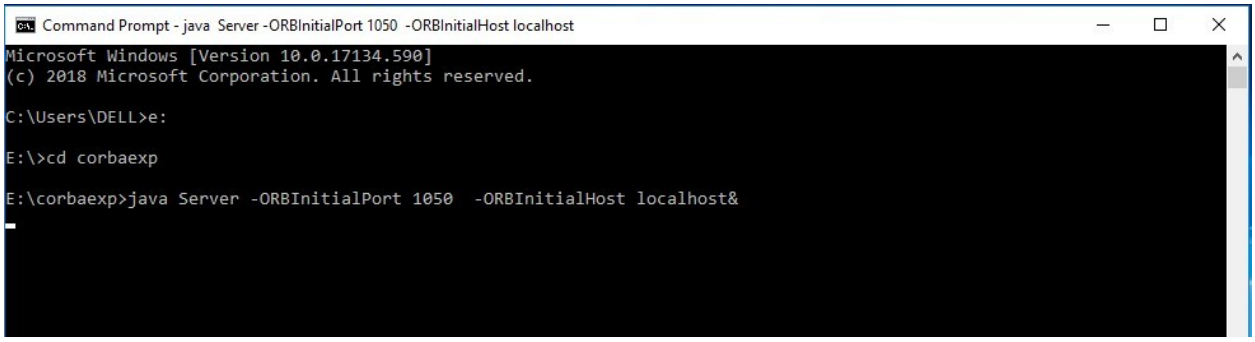
Idlj –fall –oldImplBase Arithmetic.idl
javac ArithmeticImp.java
javac Server.java javac
Client.java

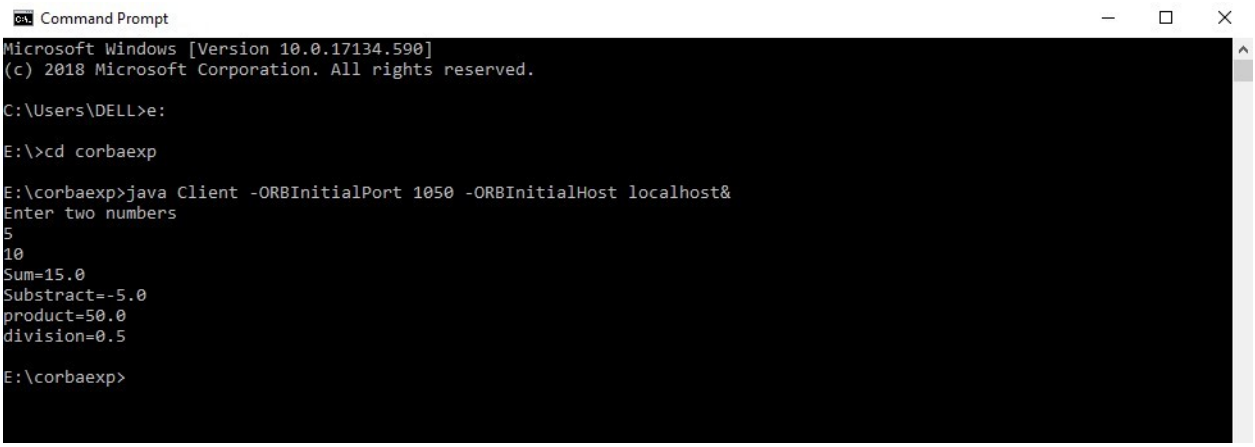
To run:

1.orbd –ORBInitialPort 1050 –ORBInitialHost localhost&
2.java Server –ORBInitialPort 1050 –ORBInitialHost localhost&
3.java Client -ORBInitialPort 1050 –ORBInitialHost localhost&

OUTPUT:




Command Prompt - java Server -ORBInitialPort 1050 -ORBInitialHost localhost
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\DELL>e:
E:\>cd corbaexp
E:\corbaexp>java Server -ORBInitialPort 1050 -ORBInitialHost localhost&


Command Prompt
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\DELL>e:
E:\>cd corbaexp
E:\corbaexp>java Client -ORBInitialPort 1050 -ORBInitialHost localhost&
Enter two numbers
5
10
Sum=15.0
Substract=-5.0
product=50.0
division=0.5
E:\corbaexp>