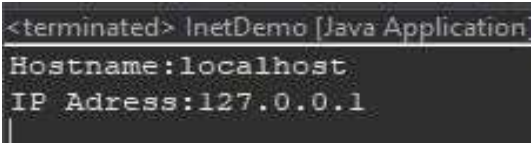


1.InetAddressDemo

Program:

```
import java.net.InetAddress; public
class InetDemo {
    public static void main(String[] args) {
        try {
            InetAddress ip=InetAddress.getByName("localhost");
            System.out.println("Hostname:"+ip.getHostName());
            System.out.println("IP Adress:"+ip.getHostAddress());
        }catch(Exception e) {
            System.out.println(e);
        }
    }
}
```

OUTPUT:



```
<terminated> InetDemo [Java Application]
Hostname:localhost
IP Adress:127.0.0.1
```

2.Client server program for translating given name into upper case

Program:

Server

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;
public class ClientServerDemo {
```

```

public static void main(String[] args) {
    int portNumber = 1341;
    try {
        ServerSocket ss = new ServerSocket(portNumber);

        System.out.println("Server is waiting for client connections on port " +
portNumber);

        while (true) {
            Socket clientSocket = ss.accept();

            System.out.println("Accepted connection from client: " +
clientSocket.getInetAddress());

            Thread clientThread = new Thread(() -> handleClient(clientSocket));
            clientThread.start();
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}

private static void handleClient(Socket clientSocket) {
    try (
        BufferedReader reader = new BufferedReader(new
            InputStreamReader(clientSocket.getInputStream()));
        PrintWriter writer = new PrintWriter(clientSocket.getOutputStream(), true)
    ) {
        String name = reader.readLine();

        System.out.println("Received name from client: " + name);

        String translatedName = name.toUpperCase();
        writer.println(translatedName);

        System.out.println("Sent translated name to client: " + translatedName);
        clientSocket.close();
    }
}

```

```

    } catch (IOException e) {
        e.printStackTrace();
    }
}
}
}

```

OUTPUT:

```

C:\Users\devak\.jdk\openjdk-21.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.3.2
Server is waiting for client connections on port 1341

```

Client

```

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;
public class ServerClientDemo {
    public static void main(String[] args) {
        {
            try (
                Socket socket = new Socket("localhost", 1341);
                BufferedReader reader = new BufferedReader(new
                    InputStreamReader(System.in));
                PrintWriter writer = new
                    PrintWriter(socket.getOutputStream(), true);
                BufferedReader serverReader = new BufferedReader(new
                    InputStreamReader(socket.getInputStream()))
            ) {

```

```

        System.out.print("Enter a name to be translated to upper case: ");

        String name = reader.readLine();

        writer.println(name);

        String translatedName = serverReader.readLine();

        System.out.println("Translated name received from server: " +
translatedName);

    } catch (IOException e) {

        e.printStackTrace();

    }

}

}

}

```

OUTPUT:

```

C:\Users\devak\.jdk\openjdk-21.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition
Enter a name to be translated to upper case:  deva
Translated name received from server: DEVA

Process finished with exit code 0

```

3.URLDemo

Program:

```

import java.net.MalformedURLException;

import java.net.URL;

public class URLEDemo {

    public static void main(String[] args) {

        try {

            URL url=new URL("https://www.greeksforgreeks.org/java");

            System.out.println(url);

            System.out.println("protcol:"+url.getProtocol());


```

```

        System.out.println("HostName:"+url.getHost());

        System.out.println("Port Number:"+url.getPort());

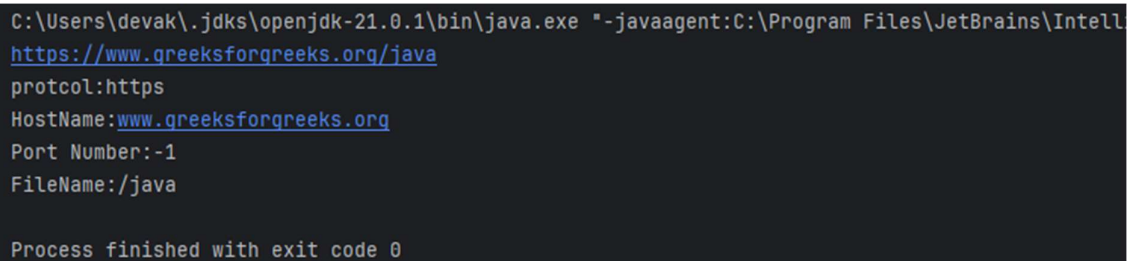
        System.out.println("FileName:"+url.getFile());
    }catch(MalformedURLException e)
    {

        System.out.println(e);

    }
}

```

OUTPUT:



```

C:\Users\devak\.jdk\openjdk-21.0.1\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\bin\idea_rt.jar -Didea.launcher.debug=https://www.greeksforgreeks.org/java
protocol:https
HostName:www.greeksforgreeks.org
Port Number:-1
FileName:/java

Process finished with exit code 0

```

4.URLConnectionDemo

Program:

```

import java.io.InputStream;

import java.net.URL;

import java.net.URLConnection;

public class URLConnectionExample {

    public static void main(String[] args){

        try

        {

            URL url=new URL("https://www.javatpoint.com/URLConnectionclass");

            URLConnection urlcon=url.openConnection();

            System.out.print(urlcon.getContentLength());

            System.out.print(urlcon.getConnectTimeout());

        }

    }

}

```


5.DatagramSockets

Program:

Sender

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;

public class DSender {

    public static void main(String[] args) throws Exception{

        DatagramSocket ds = new DatagramSocket();

        String str = "Welcome java";

        InetAddress ip = InetAddress.getByName("127.0.0.1");

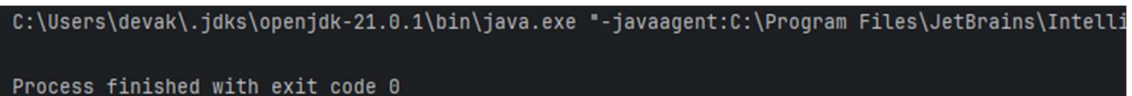
        DatagramPacket dp = new DatagramPacket(str.getBytes(),str.length(),ip,3000);
ds.send(dp);

        ds.close();

    }

}
```

OUTPUT:



```
C:\Users\devak\.jdk\openjdk-21.0.1\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ
Process finished with exit code 0
```

Receiver

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;

public class DReceiver {

    public static void main(String[] args) throws Exception {

        DatagramSocket ds = new DatagramSocket(3000);
```

```

byte[] buff= new byte[1024];

DatagramPacket dp=new DatagramPacket(buff,1024);

ds.receive(dp);

String str = new String(dp.getData(),0, dp.getLength());

System.out.println(str);

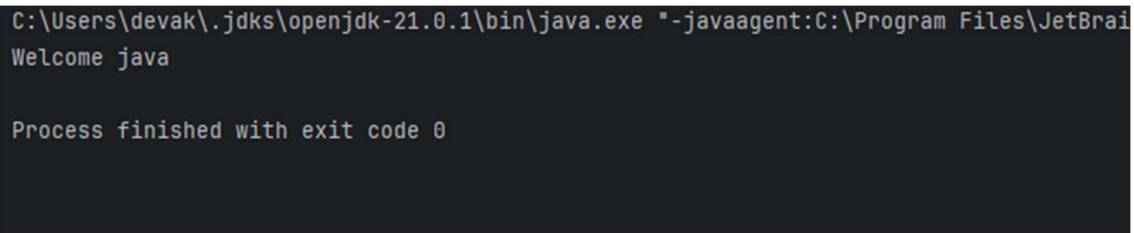
ds.close();

}

}

```

OUTPUT:



```

C:\Users\devak\.jdk\openjdk-21.0.1\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\bin\idea_rt.jar -Didea.config.path=C:\Program Files\JetBrains\IntelliJ IDEA\bin\idea.config.path java
Welcome java

Process finished with exit code 0

```

6.JDBC Program for performing DML Operations

Program

```

import java.sql.*;

import java.util.Scanner;

public class DMLOperations{

    public static void main(String[] args) {

        try (Connection con =
DriverManager.getConnection("jdbc:postgresql://localhost/postgres", "postgres",
"postgres"))

        {

            System.out.println("Connected to the PostgreSQL server.");

            Statement stmt = con.createStatement();

            int choice;

```



```

Scanner sc = new Scanner(System.in);
do {
    System.out.println("1. Insert");
    System.out.println("2. Update");
    System.out.println("3. Select");
    System.out.println("4. Delete");
    System.out.println("5. Exit");
    System.out.print("Enter your choice: ");
    choice = sc.nextInt();
    switch (choice) {
        case 1:
            String insertQuery = "INSERT INTO programmer (pname, dob, doj,
gender, prof1, prof2, salary) VALUES ('Dev', '1967-12-01', '1997-01-29', 'M', 'Pascal',
'Oracle', 4100)";
            int rowsInserted = stmt.executeUpdate(insertQuery);
            System.out.println("Rows inserted: " + rowsInserted);
            break;
        case 2:
            String updateQuery = "UPDATE programmer SET salary = 3800 WHERE
pname = 'Alice'";
            int rowsUpdated = stmt.executeUpdate(updateQuery);
            System.out.println("Rows updated: " + rowsUpdated);
            break;
        case 3:
            String selectQuery = "SELECT * FROM programmer";
            ResultSet resultSet = stmt.executeQuery(selectQuery);
            System.out.println("Programmer table:");
            while (resultSet.next()) {
                System.out.println("Name: " + resultSet.getString("pname") +

```

```

        ", Salary: " + resultSet.getDouble("salary"));
    }
    break;
case 4:
    String deleteQuery = "DELETE FROM programmer WHERE
pname='Bob'";
    int rowsDeleted = stmt.executeUpdate(deleteQuery);
    System.out.println("Rows deleted: " + rowsDeleted);
    break;
case 5:
    System.out.println("Exiting program.");
    break;
default:
    System.out.println("Invalid choice. Please enter a valid option.");
}
} while (choice != 5);
stmt.close();
} catch (SQLException e) {
    e.printStackTrace();
}
}
}

```

OUTPUT:

```

C:\Users\devak\.jdk\openjdk-21.0.1\bin\java.exe --javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.3.2\lib\idea_rt.jar=52
Connected to the PostgreSQL server.
1. Insert
2. Update
3. Select
4. Delete
5. Exit
Enter your choice: 1
Rows inserted: 1
1. Insert
2. Update
3. Select
4. Delete
5. Exit
Enter your choice: 2
Rows updated: 1
1. Insert
2. Update
3. Select
4. Delete
5. Exit
Enter your choice:

```

7.Displaying Metadata of results

Program

```

import java.sql.*;

public class ResultSetDemo {

    public static void main(String[] args) throws Exception {

        Class.forName("org.postgresql.Driver");

        Connection con =

            DriverManager.getConnection("jdbc:postgresql://localhost:5432/postgres",
"postgres", "postgres");

        Statement stmt = con.createStatement();

        ResultSet rs = stmt.executeQuery("select * from programmer");

        ResultSetMetaData rsmd = rs.getMetaData();

        int cnt = rsmd.getColumnCount();

        int i=1;

        while(rs.next()){

            while(i<cnt){

                System.out.println(rs.getObject(i++));

            }

        }

    }

}

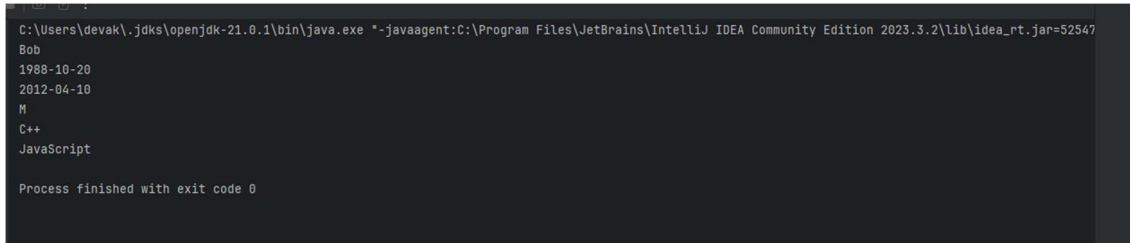
```

```

        con.close();
    }
}

```

OUTPUT:



```

C:\Users\devak\.jdk\openjdk-21.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.3.2\lib\idea_rt.jar=52547
Bob
1988-10-20
2012-04-10
M
C++
JavaScript
Process finished with exit code 0

```

8.Displaying of Metadata of DB

Program

```

import java.sql.*;

public class MetaDataDB {    public static void main(String[] args) {

    try {

        Class.forName("org.postgresql.Driver");

        Connection con =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/postgres",
"postgres", "postgres");

        DatabaseMetaData metaData = con.getMetaData();

        String tableName = "programmer";

        ResultSet resultSet = metaData.getColumns(null, null, tableName, null);

        System.out.println("Column information for table: " + tableName);

        while (resultSet.next()) {

            String columnName = resultSet.getString("COLUMN_NAME");           String
dataType = resultSet.getString("TYPE_NAME");           int columnSize =
resultSet.getInt("COLUMN_SIZE");

            System.out.println("Column Name: " + columnName);

            System.out.println("Data Type: " + dataType);

            System.out.println("Column Size: " + columnSize);

```

```

        System.out.println("-----");
    }

} catch (ClassNotFoundException | SQLException e) {
    e.printStackTrace();
}

}

}

```

OUTPUT:

```

C:\Users\devak\.jdk\openjdk-21.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.3.2\lib\idea_rt.jar=5255
Column information for table: programmer
Column Name: pname
Data Type: varchar
Column Size: 20
-----
Column Name: dob
Data Type: date
Column Size: 13
-----
Column Name: doj
Data Type: date
Column Size: 13
-----
Column Name: gender
Data Type: bpchar
Column Size: 1
-----
Column Name: prof1
Data Type: varchar
Column Size: 10
-----

```

9.PreparedstatementDemo

Program

```

import java.sql.*;

public class PreparedStatementDemo {

    public static void main(String[] args) throws Exception {

        Class.forName("org.postgresql.Driver");

        Connection con =
DriverManager.getConnection("jdbc:postgresql://localhost/postgres","postgres","pos
tgres");

        Statement st = con.createStatement();

```

```

        PreparedStatement stmt = con.prepareStatement("insert into programmer
values(?,?,?,?,?,?,?)");

        stmt.setString(1,"Pavansai");

        stmt.setDate(2,new Date(1961,12,13));

        stmt.setDate(3,new Date(1995,12,13));

        stmt.setString(4,"M");

        stmt.setString(5,"Fortran");

        stmt.setString(6,"Dbase");

        stmt.setFloat(7,3600);

        int cnt = stmt.executeUpdate();

        if(cnt!=0)

            System.out.println("Successfully inserted");

        else

            System.out.println("Not inserted");

        ResultSet rs=st.executeQuery("select * from programmer");

        while(rs.next())

            System.out.println(rs.getString(1)+" "+rs.getDate(2)+" "+rs.getDate(3)+"
"+rs.getString(4)+" "+rs.getString(5)+" "+rs.getString(6)+" "+rs.getInt(7));

    }

}

```

OUTPUT:

```

C:\Users\devak\jdk\openjdk-21.0.1\bin\java.exe --javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.3.2\lib\idea_rt.jar=5257
Successfully inserted
Bob 1988-10-20 2012-04-10 M C++ JavaScript 80000
Charlie 1995-12-28 2018-07-05 M Python Ruby 70000
Diana 1992-07-10 2016-11-30 F JavaScript HTML 85000
John 1967-12-01 1997-01-29 M Pascal Oracle 4100
Dev 1967-12-01 1997-01-29 M Pascal Oracle 4100
Alice 1990-05-15 2015-09-20 F Java Python 3800
Pavan 3862-01-13 3896-01-13 M Fortran Dbase 3600
Ravi 3863-04-13 3893-08-13 M C Assembly 4100
Pavansai 3862-01-13 3896-01-13 M Fortran Dbase 3600

```

10.CallablestatementDemo

Program

```
import java.sql.*;

public class CallableStatementDemo {

    public static void main(String[] args) throws Exception {

        Class.forName("org.postgresql.Driver");

        Connection con =
DriverManager.getConnection("jdbc:postgresql://localhost/postgres", "postgres",
"postgres");

        String sql_string = "insert into programmer values(?,?,?,?);";

        CallableStatement cs = con.prepareCall(sql_string);

        cs.setString(1,"Ravi");

        cs.setDate(2,new Date(1963,3,13));

        cs.setDate(3,new Date(1993,7,13));    cs.setString(4,"M");

        cs.setString(5,"C");    cs.setString(6,"Assembly");    cs.setFloat(7,4100);
cs.execute();

        System.out.println("Successfully Inserted");

        Statement st =

con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATA
BLE);

        ResultSet rs = st.executeQuery("select * from programmer where
pname='Ravi'");

        int cnt=rs.getMetaData().getColumnCount();    int i=1;

        System.out.println("Inserted Details are:");    while(rs.next())

        {

            while(i<=cnt){

                System.out.println(rs.getObject(i++));

            }

        }

    }

}
```

```
}  
  
}
```

OUTPUT:

```
C:\Users\devak\.jdk\openjdk-21.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.3.2\lib\idea_rt.jar=5257:  
Successfully Inserted  
Inserted Details are:  
Ravi  
3863-04-13  
3893-08-13  
M  
C  
Assembly  
4100.00  
  
Process finished with exit code 0
```