**********Networking Basics**********

- **Java NetWorking:-**

Java Networking is a concept of connecting two or more computing devices together so that we can share resources.

Java socket programming provides facility to share data between different computing devices.

- **Advantages of Java Networking:-**

1)Share Resources.
2)Centralize software Management.

- **NetWorking Terms:-**

1)IP Address:-

IP Address is a unique number assigned to a node of a network eg. 192.168.0.1,It is composed of octets that range from 0 to 255.

2)Protocol:-

A Protocol is a set of rules basically this is Followed for Communication
For Example:-

I)TCP
II)FTP
III)Telnet
IV)SMTP
V)POP ext.

3)Port Number:-

The Port Number is used to uniquely Identify different application.
It acts as a Communication endpoint between application.

The port number is associated with the IP address for Communication between two Applications.

port is a number socket on a Particular machine.

4)Connection-oriented and connection-less Protocol:-

In Connection-oriented Protocol,acknowledgement is sent by the receiver.so it is reliable but slow.The Example of Connection-oriented Protocol is TCP.

But,In Connection-less Protocol,acknowledgement is not sent by the receiver. so it is not reliable but fast.The Example of Connection-less Protocol is UDP.

5)Socket:-

A Socket is an endpoint betweentwo way Communication.

6)Proxy Servers

7)Client-Server

8)Reserved Socket

9)Internet Addressing

A-> 0 to 127 **B->** 128 to 191 **C->** 192 to 223 **D->** 224 to 239 **E->** 239 to 255.

10)Domain Naming Services.

- **#### InetAddress ####**

- java.net package
- InetAddress stands for Internet Address
- InetAddress is a combination of IP address and Host Name.
- InetAddress is a predefined class which is present under java.net package.

- ***Factory Methods:-**

1) static InetAddress getLocalHost()
2) static InetAddress getByName(String hot_name)
3) static InetAddress getAllByName(String hot_name)
4) static InetAddress getByAddress(String IP_Address)

- All these three methods will throws UnKnownHostException

- ***Instance Methods:-**
  1) boolean equals(object obj)
  2) byte[] getAddress()
  3) String getHostAddress();
  4) String getHostName();
  5) String toString();
  6) boolean isMulticastAddress()

- **Code:-**

```
import java.net.*;
class InetAddressDemo
{
    public static void main(String args[])throws UnknownHostException
    {
            InetAddress addr=InetAddress.getLocalHost();
            System.out.println(addr);

            addr=InetAddress.getByName("www.msbte.com");
            System.out.println(addr);

            System.out.println("(wwww.msbte.com)Is Multicast Addresss
            Multicast Address="+addr.isMulticastAddress());

        InetAddress a[]=InetAddress.getAllByName("www.google.com");
        for(int i=0;i<a.length;i++)
        {
            System.out.println(a[i]);
        }

    }
}
```

- **URL:-**
  - URL stands for Uniform Resource Locator.
  - Here, we need to handle the exception **MalformedURLException**.
  - URL string consists of three part
  1) Network protocol

2) Host Name or IP address
3) File path or resource location

Syntax:-

Protocol://hostnameorIPaddress:portno/filepath

Example:-

http://www.vjtechacademy.com/about.html

- ***Constructor:-
  1) URL(String urlstring)
  2) URL(String protocolName, String HostName, int port, String filepath);
  3) URL(String protocolName, String HostName,String filepath);
- ***Methods:-
  1) String getProtocol();
  2) int getPort();
  3) String getHost();
  4) String getFile();

- Code:-

```
import java.net.*;
class URLDemo
{
 public static void main(String args[])throws MalformedURLException
 {
     URL u1=new URL("https://www.vjtechacademy.com80/about.html");
            System.out.println("Protoclol="+u1.getProtocol());
            System.out.println("Host Name="+u1.getHost());
            System.out.println("Port Name="+u1.getPort());
            System.out.println("File path="+u1.getFile());
 }
}
```

- URLConnection:-
  - It is used to access attributes of remote resources.
- **Methods:-
  1) int getContentLength()
  2) String getContentType()
  3) long getDate();
  4) long getlastModified()

5) long getExpiration()
6) InputStream getInputStream()

- **Code:-**

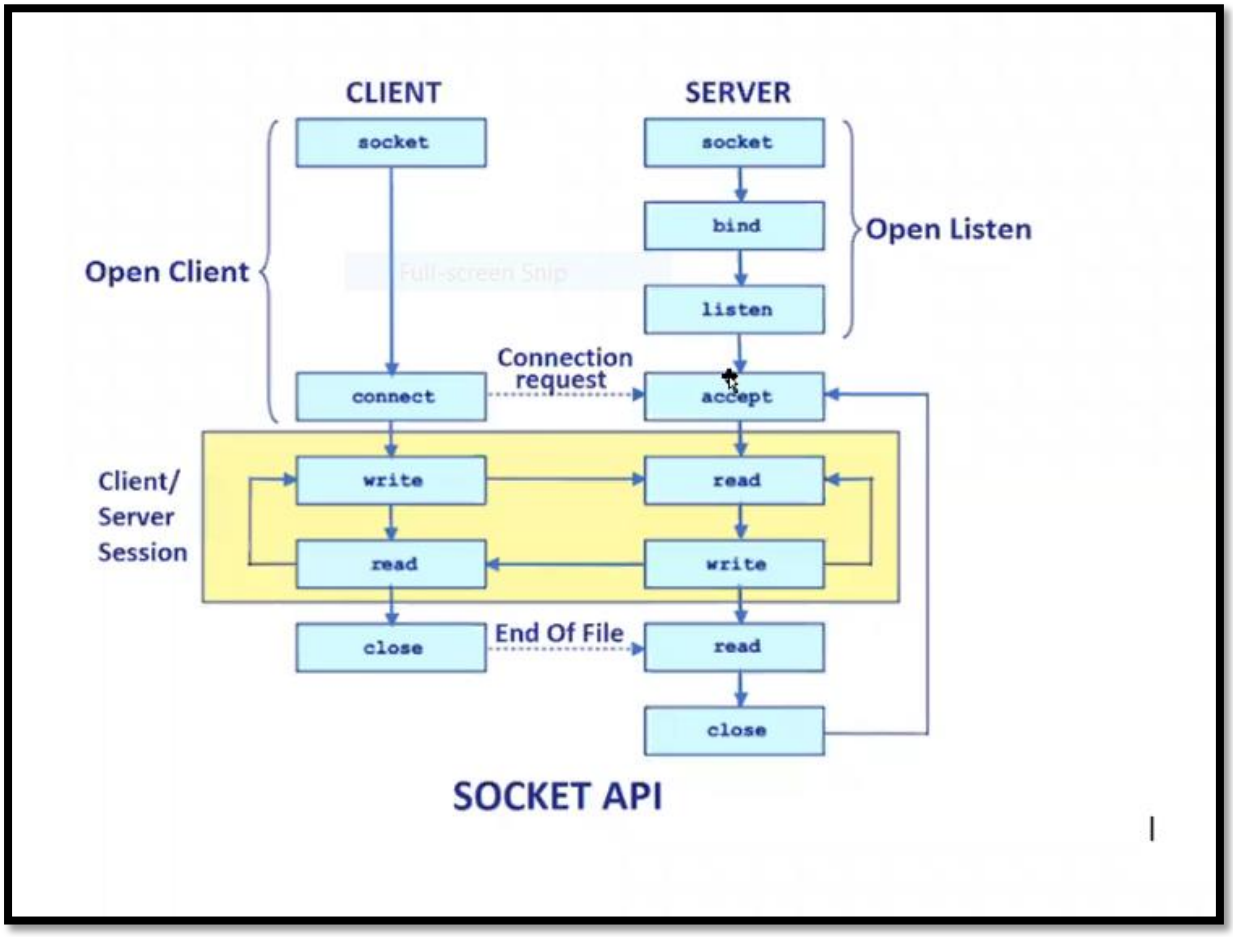
```
import java.net.*;
import java.util.*;

class URLConnectionDemo
{
  public static void main(String args[])throws Exception
  {
                URL u1=new URL("https://www.msbte.com");
                URLConnection u2=u1.openConnection();

                System.out.println("Date="+new Date(u2.getDate()));
                System.out.println("Type of contents="+(u2.getContentType()));
               System.out.println("length of contents="+(u2.getContentLength()));
              System.out.println("Expiration Date="+new Date(u2.getExpiration()));
    System.out.println("Last Modified Date="+new Date(u2.getLastModified()));
  }
}
```

➢ TCP/IP server (ServerSocket) and client (Socket):

- ServerSocket and Socket both are predefined classes which are present under java.net package.

- ***Socket class:**
- **Constructors:-**
  1) Socket(String HostName,int port)
  2) Socket(InetAddress addr,int port)
- **Methods:-**
  1) InetAddress getInetAddress();
  2) int getPort();
  3) int getLocalPort();
  4) InputStream getInputStream()
  5) OutputStream getOutputStream()
  6) void close()
- ***ServerSocket Class:-**

- **Constructors:-**
  1) ServerSocket(int port)
  2) ServerSocket(int port , int maxqueue)
  3) ServerSocket(int port,int maxqueue,InetAddress addr);
  4) accept();
- **Diagram:-**



- **Code:-**

//CLIENT Program

```
import java.net.*;
import java.io.*;
class Client
{
        public static void main(String args[])throws Exception
       {
                    Socket s=new Socket("locolhost",8989);
                    InputStream in=s.getInputStream();
                   OutputStream out=s.getOutputStream();
```

```
                    byte str[]="Hi Server".getBytes();
                    out.write(str);
                    s.close();
            }
      }
```

//SERVER Program

```
    import java.net.*;
    import java.io.*;
    class Server
    {
        public static void main(String args[])throws Exception
        {
            ServerSocket ss=new ServerSocket(8989);
            Socket s1=ss.accept();
            InputStream in=s1.getInputStream();
            OutputStream out=s1.getOutputStream();
             int c;
             while((c=in.read())!=-1)
             {
                System.out.print((char)c);
             }
             ss.close();
          }
    }
```

****

- **Code 2:-**

//Client-1 Program

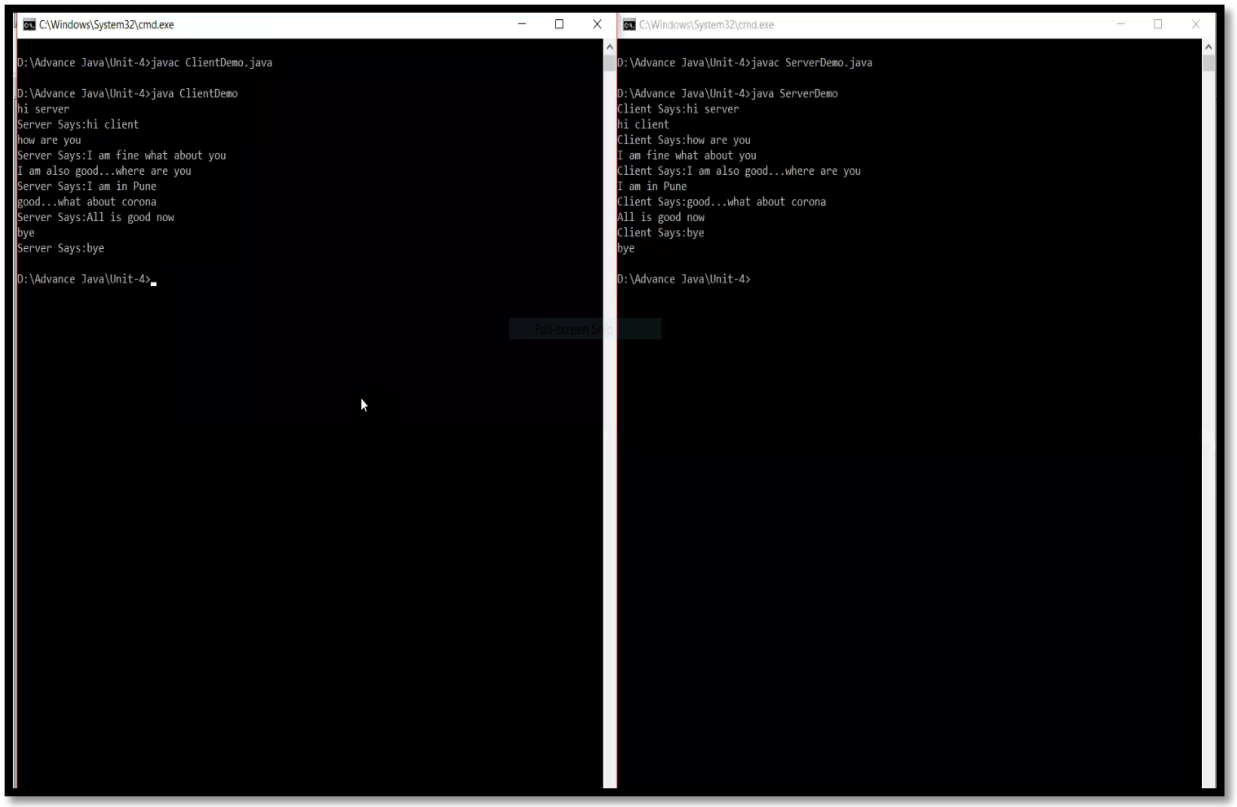
```
import java.net.*;
import java.io.*;
class ClientDemo
{
   public static void main(String args[])throws Exception
   {
      Socket s=new Socket("localhost",9090);
      BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
      DataInputStream din=new DataInputStream(s.getInputStream());
      DataOutputStream dout=new
                  DataOutputStream(s.getOutputStream());
      String str1="";
      while(!str1.equals("bye"))
```

```
        {
            str1=br.readLine();
            dout.writeUTF(str1);
            str1=din.readUTF();
            System.out.println("Server says:"+str1);
        }
        s.close();
    }
}
```

//server-1 Program

```
import java.net.*;
import java.io.*;
class ServerDemo
{
    public static void main(String args[])throws Exception
    {
            ServerSocket ss=new ServerSocket(9090);
            Socket s=ss.accept();
            BufferedReader br=new BufferedReader
                                        (new InputStreamReader(System.in));
           DataInputStream din=new DataInputStream(s.getInputStream());
            DataOutputStream dout=new DataOutputStream
                                                    (s.getOutputStream());
            String str1="";
            while(!str1.equals("bye"))
            {
                str1=din.readUTF();
                System.out.println("Client Says:"+str1);
                str1=br.readLine();
                dout.writeUTF(str1);
             }
            ss.close();
            ss.close();
      }
 }
```

- **Output:-**



****

- **Datagrams:-**

  - Here, we use two predefined classes Datagrampacket & DatagramSocket.

- ***Constructors of Datagrampacket:-**

  1) Datagrampacket(byte data[],int size)
  2) Datagrampacket(byte data[],int offset,int size)
  3) Datagrampacket(byte data[],int size,InetAddress addr,int port)
  4) Datagrampacket(byte data[],int offset, int size,InetAddress addr,int port);

- ***methods of Datagrampacket:-**

  1) InetAddress getAddress();
  2) int getPort();
  3) byte[] getData();
  4) int getLength();
  5) int getOffset();
  6) void setAddress(InetAddress addr)
  7) void setData(byte data[]);
  8) void setLength(int length)

9) void setPort(int port)

- ***Constructors of DatagramSocket:-
  1) DatagramSocket()
  2) DatagramSocket(int port)
  3) DatagramSocket(int port,InetAddress addr)
  4) DatagramSocket(DatagramSocketImpl obj)
- *** Methods of DatagramSocket:-
  1) void send(Datagrampacket obj)
  2) void receive(Datagrampacket obj)
  3) int getLocalport();
  4)int getPort();
  5)void connect(InetAddress addr,int port);
  6) void disconnect();
  7) void close();

- Code:-

//Sending DatagramPackets using DatagramSocket.

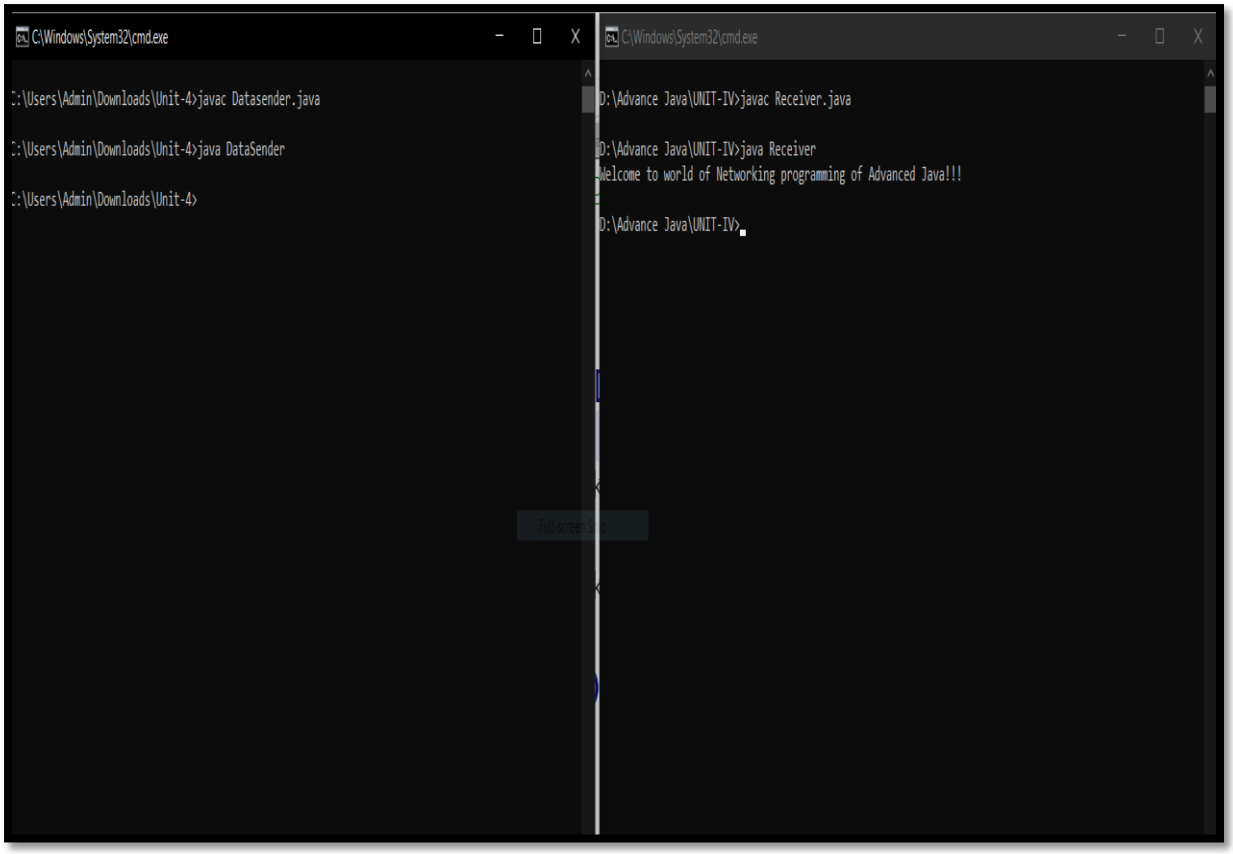
```
import java.net.*;
class DataSender
{
  public static void main(String args[])throws Exception
  {
        DatagramSocket ds=new DatagramSocket();
         String str="Welcome to world of Networking Programming
                          of Advance java!!!";
        byte data[]=str.getBytes();
        int len=str.length();
       InetAddress addr=InetAddress.getByName("127.0.0.1");
        DatagramPacket dp=new DatagramPacket(data,len,addr,5555);
        ds.send(dp);
        ds.close();
  }
}
```

//Receiving DatagramPackets using DatagramSocket.

```
import java.net.*;
class Receiver
{
```

```
  public static void main(String args[])throws Exception
  {
        DatagramSocket ds=new DatagramSocket(5555);
        byte data[]=new byte[1024];
        DatagramPacket dp=new DatagramPacket(data,1024);
        ds.receive(dp);
        String str=new String(dp.getData(),0,dp.getLength());
        System.out.println(str);
        ds.close();
  }
}
```

- **Output:-**



Inspiring Your Success

VJTech Academy…