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

public class stopWaitClient

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

 {   Socket socket = new Socket ("localhost", 4999);

   Scanner in = new Scanner (System.in);

    System.out.println ("Enter no. of frames to be sent: ");

     int n = in.nextInt ();

    for(int i=1;i<=n+1;)

{   PrintWriter prut = newPrintWriter(socket.getOutputStream());

if (i==n+1)

 {    prut. println("exit");

prut.flush();

 break;   }

System.out.println ("Frame" + i + "sent");

prut.println(i);          prut.flush();

BufferedReader br = new BufferedReader(new
InputStreamReader(socket. getInputStream()));

String str= br.readLine();

 if (str!=null)

{    System.out.println("Acknowledgement received");

i++;

  Thread.sleep(400);      }

 else

 prut.println(i);         } } }
```

```
import java.net.*;  import java.io.*;

public class stopWaitServer

{

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

   {

      String ext= "exit";

      ServerSocket ss= new ServerSocket(4999);

      Socket socket= ss.accept();

      String str= "R1";

      while(str.compareTo("exit")!=0)

      {

         Thread.sleep(400);

         BufferedReader br= new BufferedReader(new
InputStreamReader(socket.getInputStream()));

         str= br.readLine();

         if(str.compareTo(ext)==0)

            break;

         System.out.println("Frame"+str+"received");

         PrintWriter pw= new PrintWriter(socket.getOutputStream());

         pw.println("Received");

         pw.flush();

      }

      System.out.println("All frames received");

   }  }
```

```
import java.net.*;    import java.io.*;

public class Multicast_Sender

{

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

   {

      InetAddress group = InetAddress.getByName("225.2.2.2");

      MulticastSocket ms = new MulticastSocket();

      String msg = "UDPMulticasting";

      DatagramPacket packet = new DatagramPacket(msg.getBytes(),
msg.length(),group,2300);

      ms.send(packet);

      ms.close();

   }

}
```

```
import java.net.*;   import java.io.*;

public class Multicast_Receiver

{

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

   {

      InetAddress group = InetAddress.getByName("225.2.2.2");

      MulticastSocket ms = new MulticastSocket(2300);

      ms.joinGroup(group);

      byte[] data = new byte[1024];

      DatagramPacket packet = new DatagramPacket(data, data.length);

      ms.receive(packet);

      System.out.println(new String(data));

      ms.close();

   }

}
```

```
import java.net.*;
import java.io.*;
import java.util.*;
public class Calc_UDPServer
{    public static void main (String[] args) throws Exception
  {      DatagramSocket ds=new DatagramSocket(1889);
     byte num1[]=new byte[1024];
     DatagramPacket dp1=new DatagramPacket(num1,num1.length);
     ds.receive(dp1);          byte num2[]=new byte[1024];
     DatagramPacket dp2=new DatagramPacket(num2,num2.length);
     ds.receive(dp2);          byte choice[]=new byte[1024];
     DatagramPacket dp3=new DatagramPacket(choice,choice.length);
     ds.receive(dp3);        String str1=new String(dp1.getData());
     String str2=new String(dp2.getData());
     String str3=new String(dp3.getData());
     System.out.print("Number received: "+str1);
     System.out.print("Number received: "+str2);
     System.out.print("Option received: "+str3);
     int n1 = Integer.parseInt(str1.trim());
     int n2 = Integer.parseInt(str2.trim());
     int c = Integer.parseInt(str3.trim());
     int res = 0;
     switch(c)
     {    case 1:      res = n1 + n2;      break;
        case 2:        res = n1 - n2;     break;
        case 3:         res = n1 * n2;    break;
        case 4:        res = n1/n2;       break;         }
     byte b[]=String.valueOf(res).getBytes();
     InetAddress IP = InetAddress.getLocalHost();
     DatagramPacket dp4 = new DatagramPacket(b,b.length,IP,dp1.getPort());
     ds.send(dp4);
  } }
```

```
import java.net.*;
import java.io.*;
import java.util.*;
public class Calc_UDPClient
{    public static void main(String[] args) throws Exception
  {      Scanner sc=new Scanner(System.in);
     DatagramSocket ds=new DatagramSocket();
     System.out.print("Enter first integer: ");
     int i=sc.nextInt();
     System.out.print("Enter second integer: ");
     int j=sc.nextInt();
     System.out.print("1. ADD\n2. SUBTRACT\n3. MULTIPLY\n4. DIVISION\n");
     System.out.print("Enter Your Choice: ");
     int ch=sc.nextInt();
     InetAddress IP = InetAddress.getLocalHost();
     byte num1[] = String.valueOf(i).getBytes();
     DatagramPacket dp1 = new DatagramPacket(num1,num1.length,IP,1889);
     ds.send(dp1);
     byte num2[] = String.valueOf(j).getBytes();
     DatagramPacket dp2 = new DatagramPacket(num2,num2.length,IP,1889);
     ds.send(dp2);
     byte choice[] = String.valueOf(ch).getBytes();
     DatagramPacket dp3 = new
DatagramPacket(choice,choice.length,IP,1889);
     ds.send(dp3);
     byte data[]=new byte[1024];
     DatagramPacket dp4=new DatagramPacket(data,data.length);
     ds.receive(dp4);
     String str=new String(dp4.getData());
     System.out.println("Result: "+str);      }
```

```
import java.io.*;   import java.net.*;   import java.util.*;
public class TCPClient
{   public static void main (String[] args)
  {       Scanner sc= new Scanner(System.in);
    String str;
    try
    {   Socket s=new Socket ("localhost",8080);
DataOutputStream dout= new DataOutputStream (s.getOutputStream());
      do
      {     System.out.println("Enter message");
        str=sc.nextLine();
        dout.writeUTF(str);
        dout.flush();
      }
      while (!str.equals("quit"));       dout.close();       s.close();
    }     catch(Exception e)
    {       System.out.println(e);         }     }   }
```

```
import java.io.*;    import java.net.*;
public class TCPServer
{     public static void main(String[] args)
  {    String str;
    try
    {    ServerSocket ss = new ServerSocket(8080);
      Socket s=ss.accept();
      do
      {  DataInputStream dis = new DataInputStream(s.getInputStream());
        str = (String) dis.readUTF();
        System.out.println("message = " +str);
       }     while(!str.equals("quit"));
      ss.close();
    }       catch(Exception e)
    {
      System.out.println(e);
    }     }       }
```

```
import java.net.InetAddress;
import java.io.*;
public class IP_Host
{
   public static void main(String[] args)throws IOException
   {
     InetAddress addr = InetAddress.getLocalHost();
     String hostName  = addr.getHostName();
     System.out.println(addr);
     System.out.println(hostName);
   }    }
```

```
import java.net.InetAddress;
import java.net.UnknownHostException;
import java.io.*;
import java.util.*;
public class IP_Website
{    public static void main(String[] args)throws IOException
   {   Scanner sc = new Scanner(System.in);
     System.out.println("Enter an IP Address");
     String addr = sc.nextLine();
     try
     {        InetAddress ip = InetAddress.getByName(addr);
       if(ip.isReachable(5000))
         System.out.println("IP is Reachable: "+ip);
     }
     catch(Exception e)
     {       System.out.println("Exception Found: "+e);
     } }
```

```
import java.net.*;
import java.io.*;
import java.util.*;
public class UDPServer
{    public static void main (String[] args) throws Exception
  {     DatagramSocket ds=new DatagramSocket(1889);
     byte data[]=new byte[1024];
     DatagramPacket dp=new DatagramPacket(data,data.length);
     ds.receive(dp);
     String str=new String(dp.getData());
     System.out.print("Value received: "+str);
     int num=Integer.parseInt(str.trim());
     int res=num*num;
      byte b[]=String.valueOf(res).getBytes();
     InetAddress IP=InetAddress.getLocalHost();
     DatagramPacket dp1=new DatagramPacket(b,b.length,IP,dp.getPort());
     ds.send(dp1);
  } }
```

```
import java.net.*;
import java.io.*;
import java.util.*;
public class UDPClient
{      public static void main(String[] args) throws Exception
  {       Scanner sc=new Scanner(System.in);
       DatagramSocket ds=new DatagramSocket();
     System.out.print("Which integer will you like to give: ");
     int i=sc.nextInt();
     byte b[]=String.valueOf(i).getBytes();
     InetAddress IP=InetAddress.getLocalHost();
     DatagramPacket dp=new DatagramPacket(b,b.length,IP,1889);
     ds.send(dp);
     byte data[]=new byte[1024];
     DatagramPacket dp1=new DatagramPacket(data,data.length);
     ds.receive(dp1);
     String str=new String(dp1.getData());
     System.out.println("Number: "+str);
  }     }
```

```
import java.net.*;
import java.io.*;
import java.util.*;
public class UDPString_Client
{  public static void main(String[] args) throws Exception
   {     Scanner sc=new Scanner(System.in);
      DatagramSocket ds=new DatagramSocket();
      InetAddress IP=InetAddress.getLocalHost();
      while(true)
      {     System.out.println("Enter a String");
         String str = sc.nextLine();
         byte b[] = str.getBytes();
   DatagramPacket dp=new DatagramPacket(b,b.length,IP,8080);
         ds.send(dp);
         if(str.equals("Bye"))
         break;
      } }    }
```

```
import java.net.*;
import java.io.*;
import java.util.*;
public class UDPString_Server
{      public static void main(String[] args)throws IOException
   {     DatagramSocket ds = new DatagramSocket(8080);
      while(true)
      {        byte r[] = new byte[1024];
         DatagramPacket dp = new DatagramPacket(r,r.length);
         ds.receive(dp);
         String str=new String(dp.getData());
         if(str.trim().equals("Bye"))
         {          System.out.println("Bye...");
            break;
         }
         System.out.println("Data Received - "+ str);
    } }}
```