NAME : Bhavik Ransubhe
CLASS : TE (B) COMP
ROLL NO : 39055

PROBLEM STATEMENT :

Write a program using TCP socket for wired network for following (Use C/C++)
a. Say Hello to Each other
b. File transfer
c. Calculator (Arithmetic)
d. Calculator (Trigonometry)
Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode

---

1. Say Hello to Each other :-
CODE:-
CLIENT SIDE :

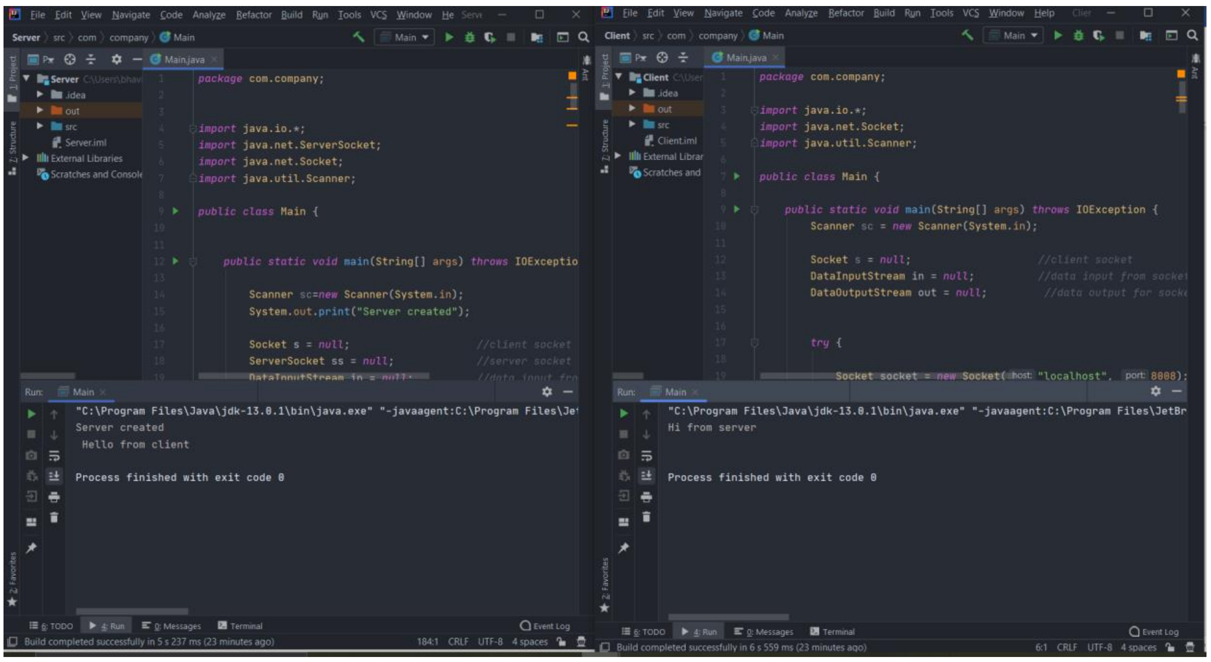
```
import java.io.*;
import java.net.Socket;
import java.util.Scanner;

public class Main {

  public static void main(String[] args) throws IOException {
    Scanner sc = new Scanner(System.in);

    Socket s = null;                //client socket
    DataInputStream in = null;        //data input from socket
    DataOutputStream out = null;       //data output for socket


    try {

      Socket socket = new Socket("localhost", 8008);
      in = new DataInputStream(socket.getInputStream());
      out = new DataOutputStream(socket.getOutputStream());


      System.out.println(in.readUTF());
      out.writeUTF("\n Hello from client");
      out.flush();

    } catch (IOException e) {
      e.printStackTrace();
    } finally {
      if (in != null) in.close();
      if (out != null) out.close();
    }
  }
}
```

SERVER SIDE:

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

```java
import java.util.Scanner;

public class Main {


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

    Scanner sc=new Scanner(System.in);
    System.out.print("Server created");

    Socket s = null;                  //client socket
    ServerSocket ss = null;           //server socket object
    DataInputStream in = null;        //data input from socket
    DataOutputStream out= null;       //data output for socket


    try {

      ss=new ServerSocket(8008);  //create serversocket with port number 8008
      s=ss.accept();
      in=new DataInputStream(s.getInputStream());
      out=new DataOutputStream(s.getOutputStream());


      out.writeUTF("Hi from server\n");   //send hi message to client
      out.flush();                        //flush all data to stream
      System.out.println(in.readUTF());      //read hi from client



    } catch (IOException e) {
      System.out.println(e);

    }finally {
      //close all allocated resource
      if(s!=null) s.close();
      if(ss!=null) ss.close();
      if(in!=null)in.close();
      if(out!=null)out.close();

    }

  }
}
```

OUTPUT:



```java
package com.company;


import java.io.*;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Scanner;

public class Main {


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

        Scanner sc=new Scanner(System.in);
        System.out.print("Server created");

        Socket s = null;                          //client socket
        ServerSocket ss = null;                   //server socket
        DataInputStream in = null;                //data input fro
```

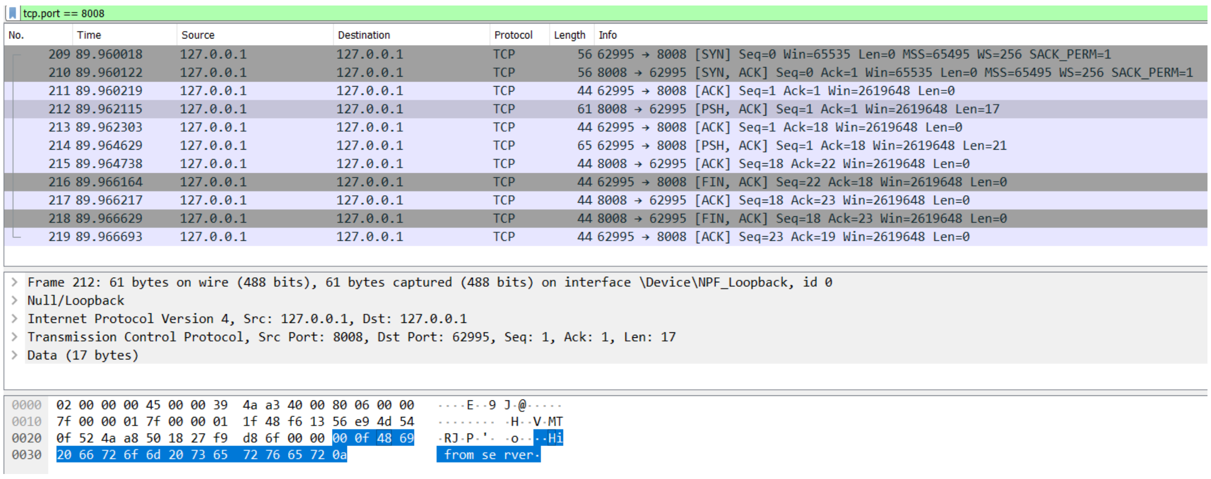
```
"C:\Program Files\Java\jdk-13.0.1\bin\java.exe" "-javaagent:C:\Program Files\Je
Server created
 Hello from client

Process finished with exit code 0
```

```java
package com.company;

import java.io.*;
import java.net.Socket;
import java.util.Scanner;

public class Main {

    public static void main(String[] args) throws IOException {
        Scanner sc = new Scanner(System.in);

        Socket s = null;                      //client socket
        DataInputStream in = null;            //data input from socket
        DataOutputStream out = null;           //data output for socke


        try {

            Socket socket = new Socket( host: "localhost",  port: 8008);
```

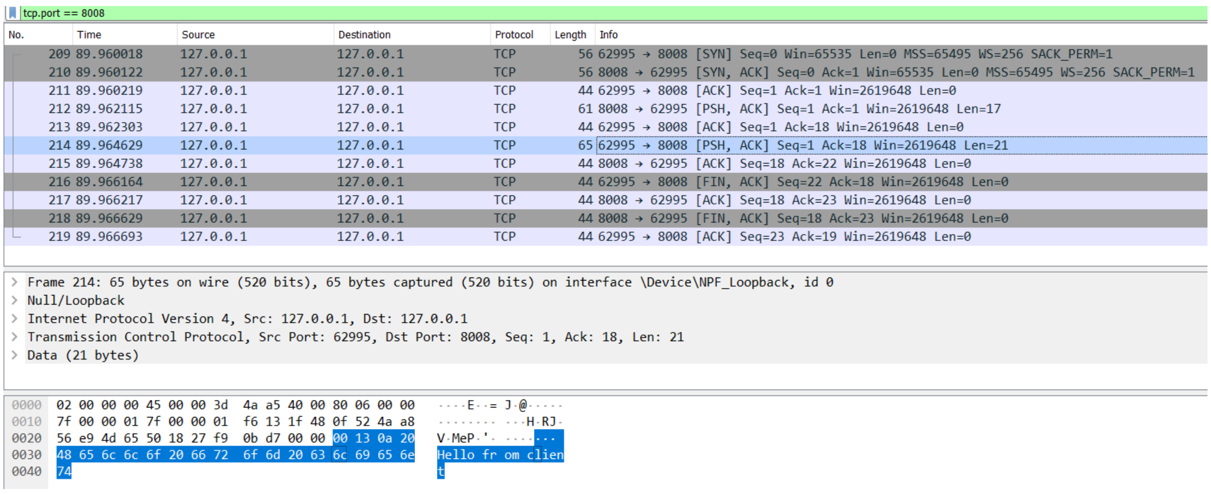
```
"C:\Program Files\Java\jdk-13.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBr
Hi from server


Process finished with exit code 0
```

Build completed successfully in 5 s 237 ms (23 minutes ago)
Build completed successfully in 6 s 559 ms (23 minutes ago)

Wireshark:



```
tcp.port == 8008

No.   Time       Source     Destination  Protocol  Length  Info
  209 89.960018  127.0.0.1  127.0.0.1    TCP       56 62995 → 8008 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
  210 89.960122  127.0.0.1  127.0.0.1    TCP       56 8008 → 62995 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
  211 89.960219  127.0.0.1  127.0.0.1    TCP       44 62995 → 8008 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
  212 89.962115  127.0.0.1  127.0.0.1    TCP       61 8008 → 62995 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=17
  213 89.962303  127.0.0.1  127.0.0.1    TCP       44 62995 → 8008 [ACK] Seq=1 Ack=18 Win=2619648 Len=0
  214 89.964629  127.0.0.1  127.0.0.1    TCP       65 62995 → 8008 [PSH, ACK] Seq=1 Ack=18 Win=2619648 Len=21
  215 89.964738  127.0.0.1  127.0.0.1    TCP       44 8008 → 62995 [ACK] Seq=18 Ack=22 Win=2619648 Len=0
  216 89.966164  127.0.0.1  127.0.0.1    TCP       44 62995 → 8008 [FIN, ACK] Seq=22 Ack=18 Win=2619648 Len=0
  217 89.966217  127.0.0.1  127.0.0.1    TCP       44 8008 → 62995 [ACK] Seq=18 Ack=23 Win=2619648 Len=0
  218 89.966629  127.0.0.1  127.0.0.1    TCP       44 8008 → 62995 [FIN, ACK] Seq=18 Ack=23 Win=2619648 Len=0
  219 89.966693  127.0.0.1  127.0.0.1    TCP       44 62995 → 8008 [ACK] Seq=23 Ack=19 Win=2619648 Len=0

> Frame 212: 61 bytes on wire (488 bits), 61 bytes captured (488 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 8008, Dst Port: 62995, Seq: 1, Ack: 1, Len: 17
> Data (17 bytes)

0000  02 00 00 00 45 00 00 39  4a a3 40 00 80 06 00 00   ····E··9 J·@·····
0010  7f 00 00 01 7f 00 00 01  1f 48 f6 13 56 e9 4d 54   ········ ·H··V·MT
0020  0f 52 4a a8 50 18 27 f9  d8 6f 00 00 00 0f 48 69   ·RJ·P·'· ·o····Hi
0030  20 66 72 6f 6d 20 73 65  72 76 65 72 0a             from se rver·
```



```
tcp.port == 8008

No.   Time       Source     Destination  Protocol  Length  Info
  209 89.960018  127.0.0.1  127.0.0.1    TCP       56 62995 → 8008 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
  210 89.960122  127.0.0.1  127.0.0.1    TCP       56 8008 → 62995 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
  211 89.960219  127.0.0.1  127.0.0.1    TCP       44 62995 → 8008 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
  212 89.962115  127.0.0.1  127.0.0.1    TCP       61 8008 → 62995 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=17
  213 89.962303  127.0.0.1  127.0.0.1    TCP       44 62995 → 8008 [ACK] Seq=1 Ack=18 Win=2619648 Len=0
  214 89.964629  127.0.0.1  127.0.0.1    TCP       65 62995 → 8008 [PSH, ACK] Seq=1 Ack=18 Win=2619648 Len=21
  215 89.964738  127.0.0.1  127.0.0.1    TCP       44 8008 → 62995 [ACK] Seq=18 Ack=22 Win=2619648 Len=0
  216 89.966164  127.0.0.1  127.0.0.1    TCP       44 62995 → 8008 [FIN, ACK] Seq=22 Ack=18 Win=2619648 Len=0
  217 89.966217  127.0.0.1  127.0.0.1    TCP       44 8008 → 62995 [ACK] Seq=18 Ack=23 Win=2619648 Len=0
  218 89.966629  127.0.0.1  127.0.0.1    TCP       44 8008 → 62995 [FIN, ACK] Seq=18 Ack=23 Win=2619648 Len=0
  219 89.966693  127.0.0.1  127.0.0.1    TCP       44 62995 → 8008 [ACK] Seq=23 Ack=19 Win=2619648 Len=0

> Frame 214: 65 bytes on wire (520 bits), 65 bytes captured (520 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 62995, Dst Port: 8008, Seq: 1, Ack: 18, Len: 21
> Data (21 bytes)

0000  02 00 00 00 45 00 00 3d  4a a5 40 00 80 06 00 00   ····E··= J·@·····
0010  7f 00 00 01 7f 00 00 01  f6 13 1f 48 0f 52 4a a8   ········ ···H·RJ·
0020  56 e9 4d 65 50 18 27 f9  0b d7 00 00 00 13 0a 20   V·MeP·'· ·····
0030  48 65 6c 6c 6f 20 66 72  6f 6d 20 63 6c 69 65 6e   Hello fr om clien
0040  74                                                  t
```

----

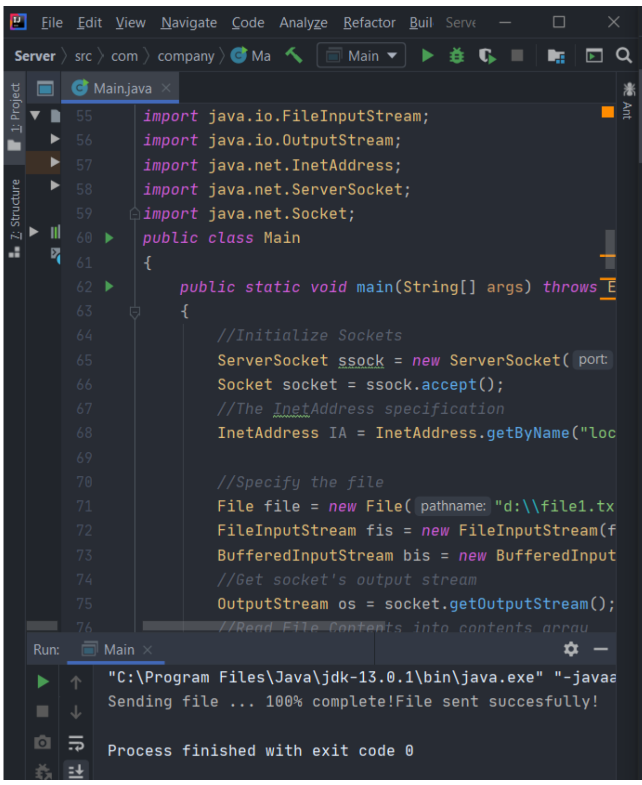
2. File transfer :-
CODE:-
CLIENT SIDE :

```java
import java.io.BufferedOutputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.net.InetAddress;
import java.net.Socket;
public class Main {
  public static void main(String[] args) throws Exception{
    //Initialize socket
    Socket socket = new Socket(InetAddress.getByName("localhost"), 5000);
    byte[] contents = new byte[10000];
    //Initialize the FileOutputStream to the output file's full path.
    FileOutputStream fos = new FileOutputStream("d:\\file2.txt");
    BufferedOutputStream bos = new BufferedOutputStream(fos);
    InputStream is = socket.getInputStream();
    //No of bytes read in one read() call
    int bytesRead = 0;
    while((bytesRead=is.read(contents))!=-1)
      bos.write(contents, 0, bytesRead);
    bos.flush();
    socket.close();
    System.out.println("File saved successfully!");
  }
}
```

SERVER SIDE:

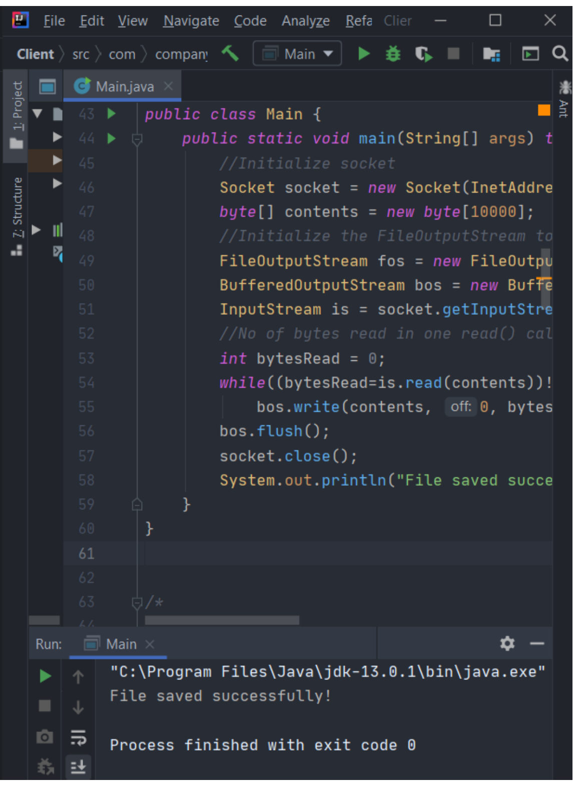
```java
import java.io.BufferedInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.OutputStream;
import java.net.InetAddress;
import java.net.ServerSocket;
import java.net.Socket;
public class Main
{
  public static void main(String[] args) throws Exception
  {
    //Initialize Sockets
    ServerSocket ssock = new ServerSocket(5000);
    Socket socket = ssock.accept();
    //The InetAddress specification
    InetAddress IA = InetAddress.getByName("localhost");

    //Specify the file
    File file = new File("d:\\file1.txt");
    FileInputStream fis = new FileInputStream(file);
    BufferedInputStream bis = new BufferedInputStream(fis);
    //Get socket's output stream
    OutputStream os = socket.getOutputStream();
    //Read File Contents into contents array
    byte[] contents;
    long fileLength = file.length();
    long current = 0;
    long start = System.nanoTime();
    while(current!=fileLength){
        int size = 10000;
        if(fileLength - current >= size)
            current += size;
        else{
            size = (int)(fileLength - current);
            current = fileLength;
        }
        contents = new byte[size];
        bis.read(contents, 0, size);
        os.write(contents);
        System.out.print("Sending file ... "+(current*100)/fileLength+"% complete!");
    }
    os.flush();
    //File transfer done. Close the socket connection!
    socket.close();
    ssock.close();
    System.out.println("File sent succesfully!");
}}
```

OUTPUT:



```java
import java.io.FileInputStream;
import java.io.OutputStream;
import java.net.InetAddress;
import java.net.ServerSocket;
import java.net.Socket;
public class Main
{
    public static void main(String[] args) throws E
    {
        //Initialize Sockets
        ServerSocket ssock = new ServerSocket( port:
        Socket socket = ssock.accept();
        //The InetAddress specification
        InetAddress IA = InetAddress.getByName("loc

        //Specify the file
        File file = new File( pathname: "d:\\file1.tx
        FileInputStream fis = new FileInputStream(f
        BufferedInputStream bis = new BufferedInput
        //Get socket's output stream
        OutputStream os = socket.getOutputStream();
        //Read File Contents into contents array
```

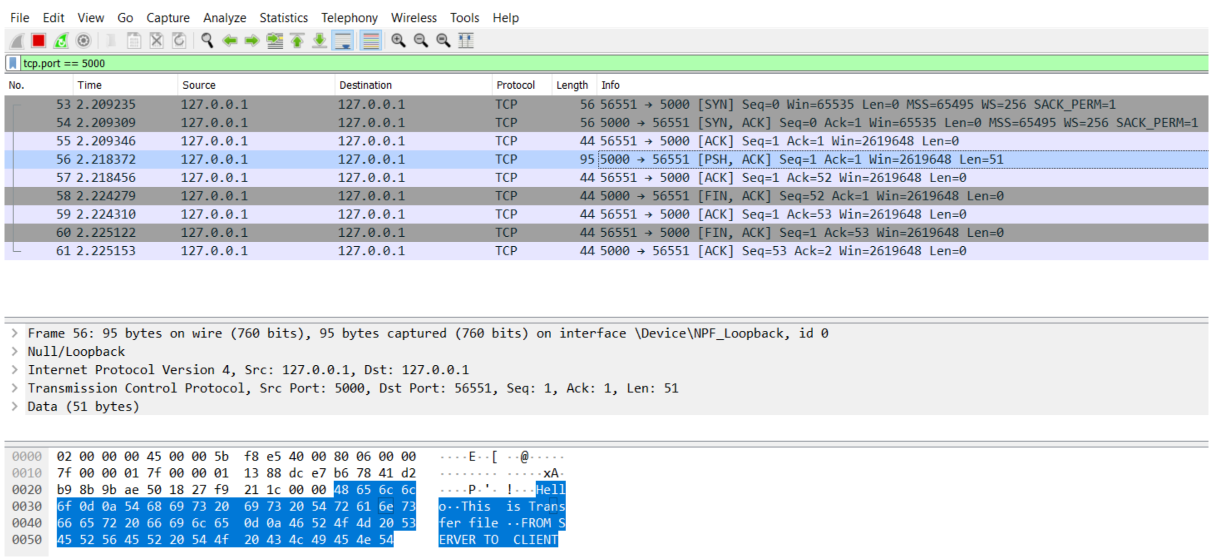
```
"C:\Program Files\Java\jdk-13.0.1\bin\java.exe" "-javaa
Sending file ... 100% complete!File sent succesfully!

Process finished with exit code 0
```



```java
public class Main {
    public static void main(String[] args) t
        //Initialize socket
        Socket socket = new Socket(InetAddre
        byte[] contents = new byte[10000];
        //Initialize the FileOutputStream to
        FileOutputStream fos = new FileOutpu
        BufferedOutputStream bos = new Buffe
        InputStream is = socket.getInputStre
        //No of bytes read in one read() cal
        int bytesRead = 0;
        while((bytesRead=is.read(contents))!
            bos.write(contents, off: 0, bytes
        bos.flush();
        socket.close();
        System.out.println("File saved succe
    }
}


/*
```

```
"C:\Program Files\Java\jdk-13.0.1\bin\java.exe"
File saved successfully!

Process finished with exit code 0
```

Wireshark:


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
tcp.port == 5000

No.   Time       Source      Destination   Protocol  Length Info
   53 2.209235   127.0.0.1   127.0.0.1     TCP       56 56551 → 5000 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
   54 2.209309   127.0.0.1   127.0.0.1     TCP       56 5000 → 56551 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
   55 2.209346   127.0.0.1   127.0.0.1     TCP       44 56551 → 5000 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
   56 2.218372   127.0.0.1   127.0.0.1     TCP       95 5000 → 56551 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=51
   57 2.218456   127.0.0.1   127.0.0.1     TCP       44 56551 → 5000 [ACK] Seq=1 Ack=52 Win=2619648 Len=0
   58 2.224279   127.0.0.1   127.0.0.1     TCP       44 5000 → 56551 [FIN, ACK] Seq=52 Ack=1 Win=2619648 Len=0
   59 2.224310   127.0.0.1   127.0.0.1     TCP       44 56551 → 5000 [ACK] Seq=1 Ack=53 Win=2619648 Len=0
   60 2.225122   127.0.0.1   127.0.0.1     TCP       44 56551 → 5000 [FIN, ACK] Seq=1 Ack=53 Win=2619648 Len=0
   61 2.225153   127.0.0.1   127.0.0.1     TCP       44 5000 → 56551 [ACK] Seq=53 Ack=2 Win=2619648 Len=0

> Frame 56: 95 bytes on wire (760 bits), 95 bytes captured (760 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 5000, Dst Port: 56551, Seq: 1, Ack: 1, Len: 51
> Data (51 bytes)

0000  02 00 00 00 45 00 00 5b  f8 e5 40 00 80 06 00 00   ····E··[ ··@·····
0010  7f 00 00 01 7f 00 00 01  13 88 dc e7 b6 78 41 d2   ········ ·····xA·
0020  b9 8b 9b ae 50 18 27 f9  21 1c 00 00 48 65 6c 6c   ····P·'· !···Hell
0030  6f 0d 0a 54 68 69 73 20  69 73 20 54 72 61 6e 73   o··This  is Trans
0040  66 65 72 20 66 69 6c 65  0d 0a 46 52 4f 4d 20 53   fer file ··FROM S
0050  45 52 56 45 52 20 54 4f  20 43 4c 49 45 4e 54      ERVER TO  CLIENT
```

---

3. Calculator (Arithmetic):-

CODE:-

CLIENT SIDE :

```java
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.InetAddress;
import java.net.Socket;
import java.util.Scanner;

public class Main
{
   public static void main(String[] args) throws IOException
   {
      InetAddress ip = InetAddress.getLocalHost();
      int port = 4444;
      Scanner sc = new Scanner(System.in);

      // Step 1: Open the socket connection.
      Socket s = new Socket(ip, port);

      // Step 2: Communication-get the input and output stream
      DataInputStream dis = new DataInputStream(s.getInputStream());
      DataOutputStream dos = new DataOutputStream(s.getOutputStream());

      while (true)
      {
         // Enter the equation in the form-
         // "operand1 operation operand2"
         System.out.print("Enter the equation in the form: ");
         System.out.println("'operand operator operand'");

         String inp = sc.nextLine();

         if (inp.equals("bye"))
            break;

         // send the equation to server
         dos.writeUTF(inp);
```

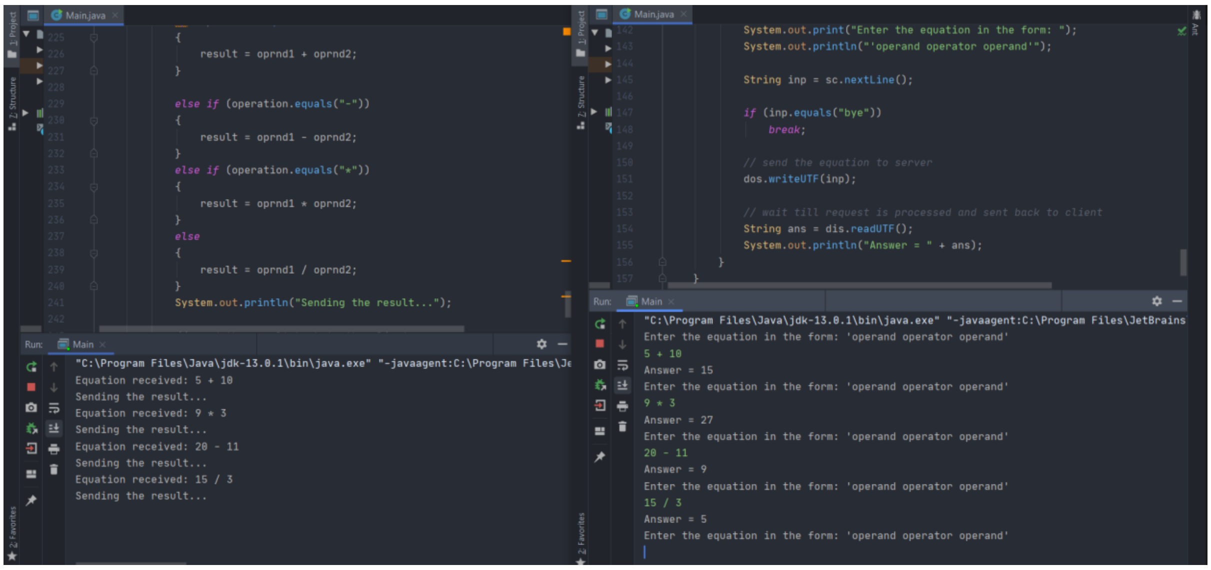
```
        // wait till request is processed and sent back to client
        String ans = dis.readUTF();
        System.out.println("Answer = " + ans);
      }
    }
}
```

SERVER SIDE:

```
import java.io.DataInputStream;
      import java.io.DataOutputStream;
      import java.io.IOException;
      import java.net.ServerSocket;
      import java.net.Socket;
      import java.util.StringTokenizer;

public class Main
{
   public static void main(String args[]) throws IOException
   {

      // Step 1: Establish the socket connection.
      ServerSocket ss = new ServerSocket(4444);
      Socket s = ss.accept();

      // Step 2: Processing the request.
      DataInputStream dis = new DataInputStream(s.getInputStream());
      DataOutputStream dos = new DataOutputStream(s.getOutputStream());

      while (true)
      {
         // wait for input
         String input = dis.readUTF();

         if(input.equals("bye"))
            break;

         System.out.println("Equation received: " + input);
         int result;

         // Use StringTokenizer to break the equation into operand and
         // operation
         StringTokenizer st = new StringTokenizer(input);

         int oprnd1 = Integer.parseInt(st.nextToken());
         String operation = st.nextToken();
         int oprnd2 = Integer.parseInt(st.nextToken());

         // perform the required operation.
         if (operation.equals("+"))
         {
            result = oprnd1 + oprnd2;
         }

         else if (operation.equals("-"))
         {
            result = oprnd1 - oprnd2;
         }
         else if (operation.equals("*"))
```

```java
            {
                result = oprnd1 * oprnd2;
            }
            else
            {
                result = oprnd1 / oprnd2;
            }
            System.out.println("Sending the result...");

            // send the result back to the client.
            dos.writeUTF(Integer.toString(result));
        }
    }
}
```

OUTPUT:



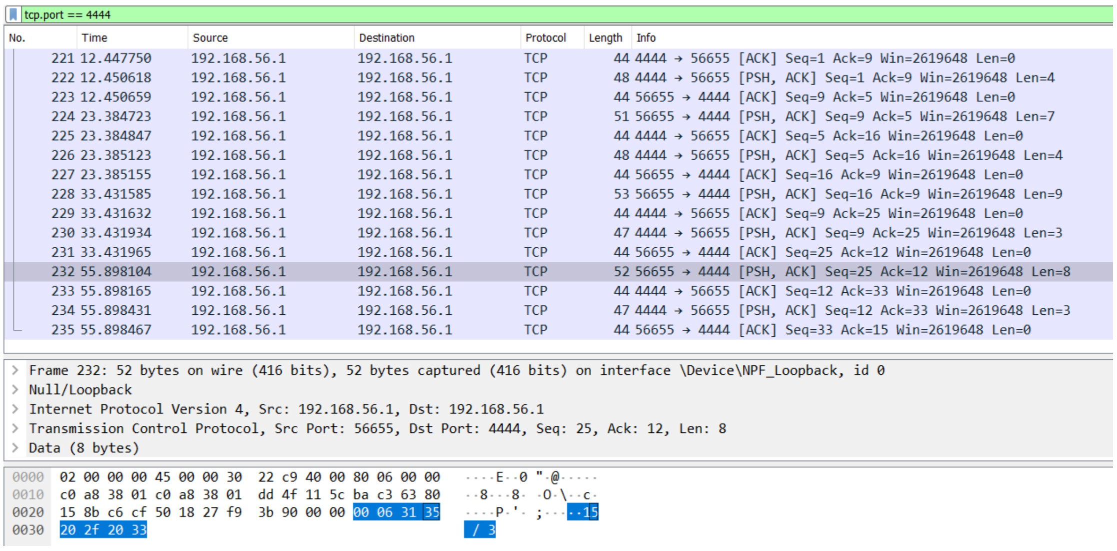
```java
        {
            result = oprnd1 + oprnd2;
        }

        else if (operation.equals("-"))
        {
            result = oprnd1 - oprnd2;
        }
        else if (operation.equals("*"))
        {
            result = oprnd1 * oprnd2;
        }
        else
        {
            result = oprnd1 / oprnd2;
        }
        System.out.println("Sending the result...");
```

```
"C:\Program Files\Java\jdk-13.0.1\bin\java.exe" "-javaagent:C:\Program Files\Je
Equation received: 5 + 10
Sending the result...
Equation received: 9 * 3
Sending the result...
Equation received: 20 - 11
Sending the result...
Equation received: 15 / 3
Sending the result...
```

```java
            System.out.print("Enter the equation in the form: ");
            System.out.println("'operand operator operand'");

            String inp = sc.nextLine();

            if (inp.equals("bye"))
                break;

            // send the equation to server
            dos.writeUTF(inp);

            // wait till request is processed and sent back to client
            String ans = dis.readUTF();
            System.out.println("Answer = " + ans);
        }
    }
```

```
"C:\Program Files\Java\jdk-13.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains
Enter the equation in the form: 'operand operator operand'
5 + 10
Answer = 15
Enter the equation in the form: 'operand operator operand'
9 * 3
Answer = 27
Enter the equation in the form: 'operand operator operand'
20 - 11
Answer = 9
Enter the equation in the form: 'operand operator operand'
15 / 3
Answer = 5
Enter the equation in the form: 'operand operator operand'
```

Wireshark:

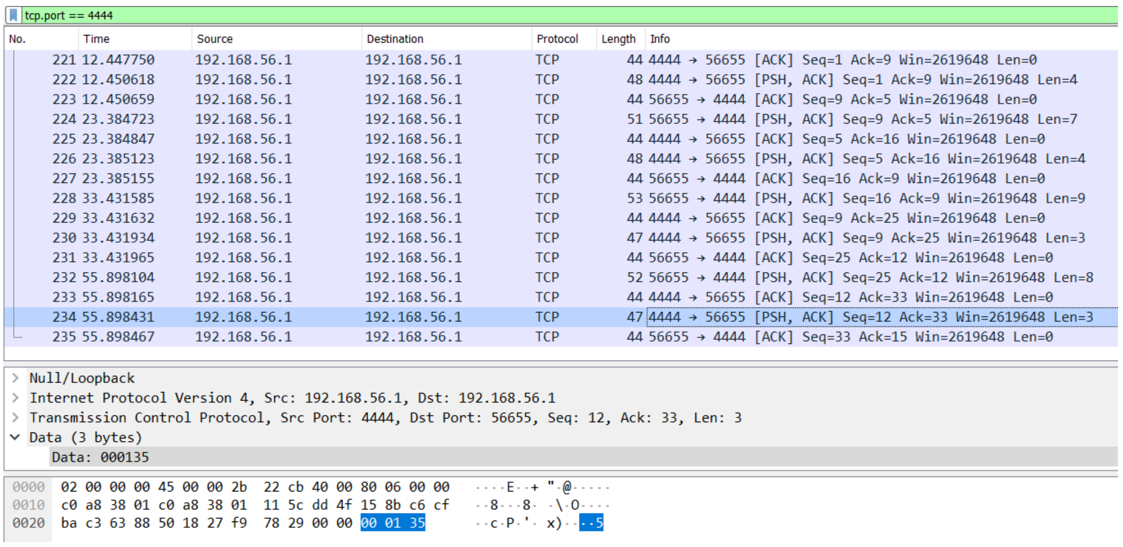


tcp.port == 4444

No.	Time	Source	Destination	Protocol	Length	Info
221	12.447750	192.168.56.1	192.168.56.1	TCP	44	4444 → 56655 [ACK] Seq=1 Ack=9 Win=2619648 Len=0
222	12.450618	192.168.56.1	192.168.56.1	TCP	48	4444 → 56655 [PSH, ACK] Seq=1 Ack=9 Win=2619648 Len=4
223	12.450659	192.168.56.1	192.168.56.1	TCP	44	56655 → 4444 [ACK] Seq=9 Ack=5 Win=2619648 Len=0
224	23.384723	192.168.56.1	192.168.56.1	TCP	51	56655 → 4444 [PSH, ACK] Seq=9 Ack=5 Win=2619648 Len=7
225	23.384847	192.168.56.1	192.168.56.1	TCP	44	4444 → 56655 [ACK] Seq=5 Ack=16 Win=2619648 Len=0
226	23.385123	192.168.56.1	192.168.56.1	TCP	48	4444 → 56655 [PSH, ACK] Seq=5 Ack=16 Win=2619648 Len=4
227	23.385155	192.168.56.1	192.168.56.1	TCP	44	56655 → 4444 [ACK] Seq=16 Ack=9 Win=2619648 Len=0
228	33.431585	192.168.56.1	192.168.56.1	TCP	53	56655 → 4444 [PSH, ACK] Seq=16 Ack=9 Win=2619648 Len=9
229	33.431632	192.168.56.1	192.168.56.1	TCP	44	4444 → 56655 [ACK] Seq=9 Ack=25 Win=2619648 Len=0
230	33.431934	192.168.56.1	192.168.56.1	TCP	47	4444 → 56655 [PSH, ACK] Seq=9 Ack=25 Win=2619648 Len=3
231	33.431965	192.168.56.1	192.168.56.1	TCP	44	56655 → 4444 [ACK] Seq=25 Ack=12 Win=2619648 Len=0
232	55.898104	192.168.56.1	192.168.56.1	TCP	52	56655 → 4444 [PSH, ACK] Seq=25 Ack=12 Win=2619648 Len=8
233	55.898165	192.168.56.1	192.168.56.1	TCP	44	4444 → 56655 [ACK] Seq=12 Ack=33 Win=2619648 Len=0
234	55.898431	192.168.56.1	192.168.56.1	TCP	47	4444 → 56655 [PSH, ACK] Seq=12 Ack=33 Win=2619648 Len=3
235	55.898467	192.168.56.1	192.168.56.1	TCP	44	56655 → 4444 [ACK] Seq=33 Ack=15 Win=2619648 Len=0

```
> Frame 232: 52 bytes on wire (416 bits), 52 bytes captured (416 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1
> Transmission Control Protocol, Src Port: 56655, Dst Port: 4444, Seq: 25, Ack: 12, Len: 8
> Data (8 bytes)
```

```
0000  02 00 00 00 45 00 00 30  22 c9 40 00 80 06 00 00   ····E··0 "·@·····
0010  c0 a8 38 01 c0 a8 38 01  dd 4f 11 5c ba c3 63 80   ··8···8· ·O·\··c·
0020  15 8b c6 cf 50 18 27 f9  3b 90 00 00 00 06 31 35   ····P·'· ;·····15
0030  20 2f 20 33                                         / 3
```



```
tcp.port == 4444
No.     Time        Source          Destination     Protocol  Length  Info
    221 12.447750   192.168.56.1    192.168.56.1    TCP       44 4444 → 56655 [ACK] Seq=1 Ack=9 Win=2619648 Len=0
    222 12.450618   192.168.56.1    192.168.56.1    TCP       48 4444 → 56655 [PSH, ACK] Seq=1 Ack=9 Win=2619648 Len=4
    223 12.450659   192.168.56.1    192.168.56.1    TCP       44 56655 → 4444 [ACK] Seq=9 Ack=5 Win=2619648 Len=0
    224 23.384723   192.168.56.1    192.168.56.1    TCP       51 56655 → 4444 [PSH, ACK] Seq=9 Ack=5 Win=2619648 Len=7
    225 23.384847   192.168.56.1    192.168.56.1    TCP       44 4444 → 56655 [ACK] Seq=5 Ack=16 Win=2619648 Len=0
    226 23.385123   192.168.56.1    192.168.56.1    TCP       48 4444 → 56655 [PSH, ACK] Seq=5 Ack=16 Win=2619648 Len=4
    227 23.385155   192.168.56.1    192.168.56.1    TCP       44 56655 → 4444 [ACK] Seq=16 Ack=9 Win=2619648 Len=0
    228 33.431585   192.168.56.1    192.168.56.1    TCP       53 56655 → 4444 [PSH, ACK] Seq=16 Ack=9 Win=2619648 Len=9
    229 33.431632   192.168.56.1    192.168.56.1    TCP       44 4444 → 56655 [ACK] Seq=9 Ack=25 Win=2619648 Len=0
    230 33.431934   192.168.56.1    192.168.56.1    TCP       47 4444 → 56655 [PSH, ACK] Seq=9 Ack=25 Win=2619648 Len=3
    231 33.431965   192.168.56.1    192.168.56.1    TCP       44 56655 → 4444 [ACK] Seq=25 Ack=12 Win=2619648 Len=0
    232 55.898104   192.168.56.1    192.168.56.1    TCP       52 56655 → 4444 [PSH, ACK] Seq=25 Ack=12 Win=2619648 Len=8
    233 55.898165   192.168.56.1    192.168.56.1    TCP       44 4444 → 56655 [ACK] Seq=12 Ack=33 Win=2619648 Len=0
    234 55.898431   192.168.56.1    192.168.56.1    TCP       47 4444 → 56655 [PSH, ACK] Seq=12 Ack=33 Win=2619648 Len=3
    235 55.898467   192.168.56.1    192.168.56.1    TCP       44 56655 → 4444 [ACK] Seq=33 Ack=15 Win=2619648 Len=0

> Null/Loopback
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.1
> Transmission Control Protocol, Src Port: 4444, Dst Port: 56655, Seq: 12, Ack: 33, Len: 3
v Data (3 bytes)
    Data: 000135

0000  02 00 00 00 45 00 00 2b  22 cb 40 00 80 06 00 00   ····E··+ "·@·····
0010  c0 a8 38 01 c0 a8 38 01  11 5c dd 4f 15 8b c6 cf   ··8···8· ·\·O····
0020  ba c3 63 88 50 18 27 f9  78 29 00 00 00 01 35      ··c·P·'· x)····5
```

4.Calculator (Trigonometry) :
CODE:-
CLIENT SIDE :

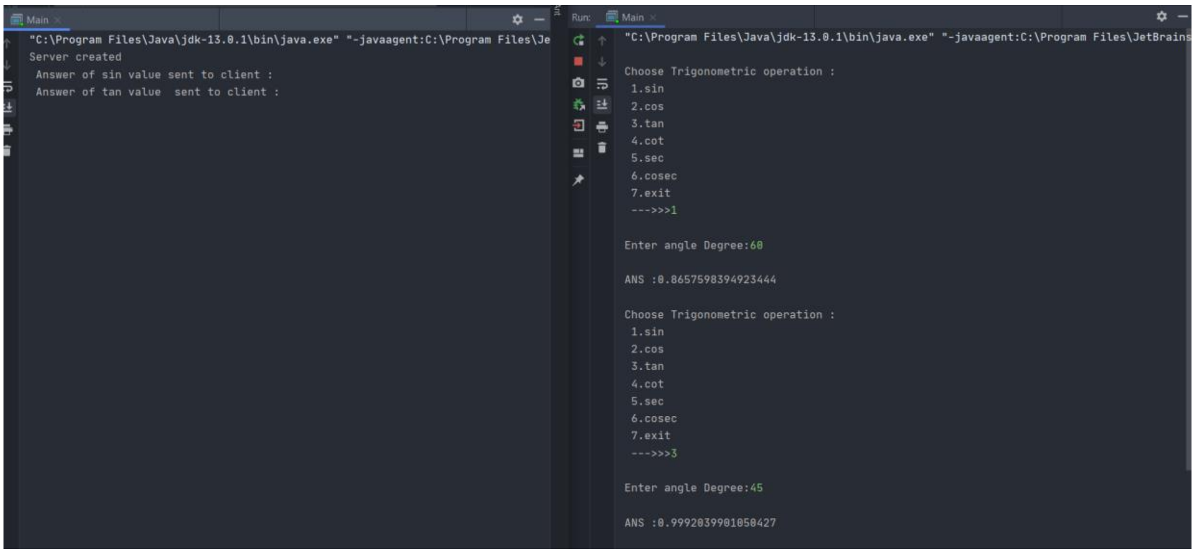
```java
import java.io.*;
import java.net.Socket;
import java.util.Scanner;

public class Main {

  public static void main(String[] args) throws IOException {
    Scanner sc = new Scanner(System.in);

    Socket s = null;                //client socket
    DataInputStream in = null;        //data input from socket
    DataOutputStream out = null;        //data output for socket

    try {

      Socket socket = new Socket("localhost", 8008);
      in = new DataInputStream(socket.getInputStream());
      out = new DataOutputStream(socket.getOutputStream());

      while (true) {

        System.out.print("\nChoose Trigonometric operation :\n 1.sin\n 2.cos\n 3.tan\n 4.cot" +
            "\n 5.sec\n 6.cosec\n 7.exit\n --->>>");
        int choice = sc.nextInt();

        if (choice < 7) {

          System.out.print("\nEnter angle Degree:");
          Double value = sc.nextDouble();
          out.writeInt(choice);
          out.writeDouble((Double) (value * 3.14 / 180)); //convert degree to radian
          System.out.println("\nANS :" + in.readDouble());  //print ans from server

        } else {
                out.writeInt('0');  //for end connection send y to server
                sc.close();       //close all allocated resources
                in.close();
                out.close();
                System.exit(0);  //exit program
            }
        }
    } catch (IOException e) {
        e.printStackTrace();
    } finally {

        if (s != null) s.close();
        if (in != null) in.close();
        if (out != null) out.close();
    }
  }
}
```

SERVER SIDE:

```

import java.io.*;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Scanner;


public class Main {


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

    Scanner sc=new Scanner(System.in);
    System.out.print("Server created");

    Socket s = null;                //client socket
    ServerSocket ss = null;           //server socket object
    DataInputStream in = null;        //data input from socket
    DataOutputStream out= null;       //data output for socket


    try {

        ss=new ServerSocket(8008);  //create serversocket with port number 8008
        s=ss.accept();
        in=new DataInputStream(s.getInputStream());
        out=new DataOutputStream(s.getOutputStream());

        int choice=in.readInt();        //read operation choice from client
        while(choice!='0')
        {
            out.writeDouble(Calculation(choice,in.readDouble()));
            choice=in.readInt();
        }

    } catch (IOException e) {
        System.out.println(e);
```

```java
        }finally {
            //close all allocated resource
            if(s!=null) s.close();
            if(ss!=null) ss.close();
            if(in!=null)in.close();
            if(out!=null)out.close();

        }

    }
    //trignometric calculation
    static Double Calculation(int choice,Double value)
    {

        switch (choice)
        {
            case 1:
                System.out.print("\n Answer of sin value sent to client :");
                return Math.sin(value);
            case 2:
                System.out.print("\n Answer of cos value sent to client :");
                return Math.cos(value);

            case 3:
                System.out.print("\n Answer of tan value  sent to client :");
                return Math.tan(value);
            case 4:
                System.out.print("\n Answer of cot value  sent to client :");
                return 1/Math.tan(value);
            case 5:
                System.out.print("\n Answer of sec value sent to client :");
                return 1/Math.cos(value);

            case 6:
                System.out.print("\n Answer of cosec value sent to client :");
                return 1/Math.sin(value);
        }
        return -1.0;
    }
}
```

OUTPUT:



```
"C:\Program Files\Java\jdk-13.0.1\bin\java.exe" "-javaagent:C:\Program Files\Je
Server created
 Answer of sin value sent to client :
 Answer of tan value  sent to client :
```

```
"C:\Program Files\Java\jdk-13.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains

Choose Trigonometric operation :
 1.sin
 2.cos
 3.tan
 4.cot
 5.sec
 6.cosec
 7.exit
 --->>>1

Enter angle Degree:60

ANS :0.8657598394923444

Choose Trigonometric operation :
 1.sin
 2.cos
 3.tan
 4.cot
 5.sec
 6.cosec
 7.exit
 --->>>3

Enter angle Degree:45

ANS :0.9992039901050427
```

****
****