Socket Programming

Problem Definition

Write a Java program using socket programming where client selects the operations and server selects the operands to do calculations.

Algorithm

Server

```
Start server on port 6666
Accept client connection

Initialize:
    - Input stream from client
    - Output stream to client
    - Console input from server user

LOOP forever
    Read operation code from client
    Prompt and read two numbers from server console

    Perform operation based on code:
        1 → Add, 2 → Subtract, 3 → Multiply, 4 → Divide
        If division by zero → send error message
        If invalid code → send invalid choice message

    Send result or message to client
END LOOP

Close all streams and sockets
```

Client

```
Get local host address
Connect to server on port 6666

Set up:
    - Input from user (console)
    - Input/output stream to server

LOOP until user enters 0
    Show operation menu
    Read user choice
    IF choice is 0 → Exit loop
    Send choice to server
    Receive result from server
    Display result
END LOOP

Close all streams and socket
```

Implementation/Code

Server

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

public class Calserver {
    public static void main(String[]
args) throws IOException {
        ServerSocket server = new
ServerSocket(6666);
        System.out.println("Server is
running... Waiting for a client to
connect...");
        Socket sock =
server.accept();
        System.out.println("Client
connected.");

        DataInputStream inpStrm = new
DataInputStream(sock.getInputStream()
);
        DataOutputStream outpStrm =
new
DataOutputStream(sock.getOutputStream
());
        Scanner sc = new
Scanner(System.in);

        try {
            while (true) {
                int oprtr =
inpStrm.readInt();

System.out.println("Client has
requested operation: " + oprtr);

System.out.print("Enter first number:
");
                int data1 =
sc.nextInt();

System.out.print("Enter second
number: ");
                int data2 =
sc.nextInt();
```

Client

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

public class Calclient {
    public static void main(String[]
args) throws IOException {
        InetAddress addr =
InetAddress.getLocalHost();
        Scanner inp = new
Scanner(System.in);
        Socket sock = new
Socket(addr, 6666);
        DataInputStream inpStrm = new
DataInputStream(sock.getInputStream()
);
        DataOutputStream outpStrm =
new
DataOutputStream(sock.getOutputStream
());

        try {
            while (true) {

System.out.println("Type 1 for
Addition");

System.out.println("Type 2 for
Subtraction");

System.out.println("Type 3 for
Multiplication");

System.out.println("Type 4 for
Division");

System.out.println("Enter 0 to
Exit");

System.out.print("Enter your choice:
");

                int oprtr =
inp.nextInt();
```

```
                if (oprtr == 0) {
                    Break;
                   case 1:
                        res = data1 +
data2;

outpStrm.writeUTF(Integer.toString(re
s));
                        break;
                    case 2:
                        res = data1 -
data2;

outpStrm.writeUTF(Integer.toString(re
s));
                        break;
                    case 3:
                        res = data1 *
data2;

outpStrm.writeUTF(Integer.toString(re
s));
                        break;
                    case 4:
                        if (data2 ==
0) {

outpStrm.writeUTF("Error: Division by
zero");
                        } else {
                            res =
data1 / data2;

outpStrm.writeUTF(Integer.toString(re
s));
                        }
                        break;
                    default:

outpStrm.writeUTF("You have given an
invalid choice!");
                        break;
                }


System.out.println("Result sent to
the client...");
            }
        } catch (Exception exp) {
                }


outpStrm.writeInt(oprtr);
                String res =
inpStrm.readUTF();

System.out.println("Your Result for
the given operation = " + res);
            }
        } catch (Exception exp) {
            System.out.println("An
error occurred: " +
exp.getMessage());
        } finally {
            inp.close();
            inpStrm.close();
            outpStrm.close();
            sock.close();
        }
    }
}
```