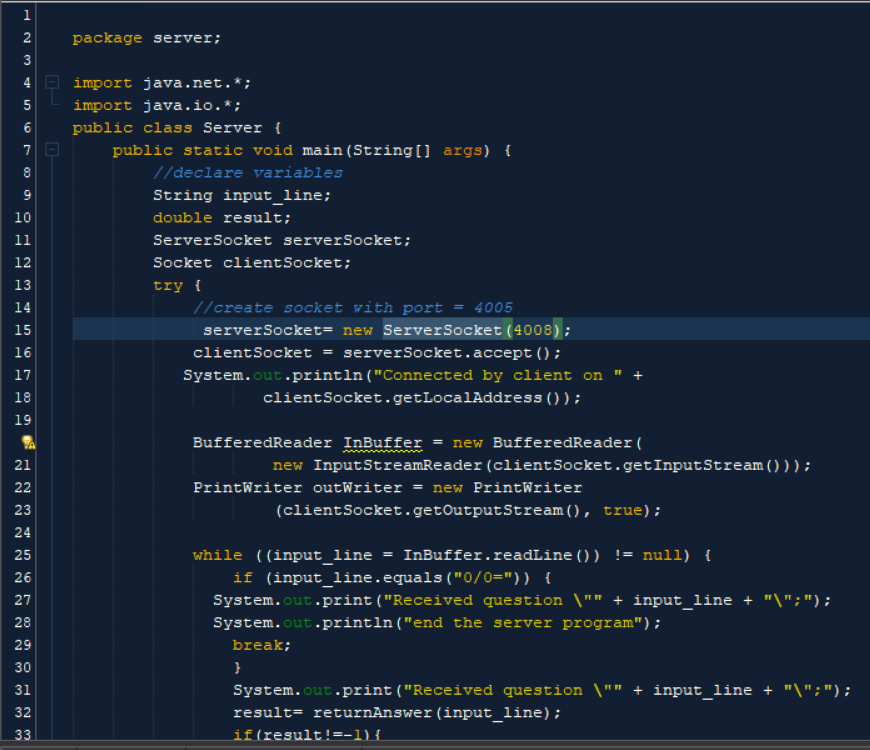
الاسم :

رقم الشعبة :

اسم الدكتور :

**Programming Language Used in this Project 🡪 Java**

**And IDE we Use is 🡪 Apache Netbeans**

* Server Code Screenshots

**Code Description**

A computer screen shot of a program

Description automatically generated

A computer screen shot of a program code

Description automatically generated

* Server Code description

1. Socket Setup:

• The server creates a ServerSocket object and binds it to port 4008 to listen for incoming connections.

• When a client connects, the server accepts the connection and creates a Socket object to communicate with the client.

2. Input and Output Streams:

• The server sets up input and output streams to communicate with the client. It reads input from the client using a BufferedReader and sends output using a PrintWriter.

3. Processing Requests:

• The server enters a loop where it continuously reads input from the client.

• If the client sends "0/0=", indicating the end of input, the server breaks out of the loop and ends its execution.

• Otherwise, it processes the received mathematical expression using the returnAnswer method and sends back the result to the client.

4. Error Handling:

• The server handles IOExceptions that might occur during socket communication. It prints error messages if any exception occurs.

Methods:

1. Main Method (main):

• The entry point of the program.

• It initializes variables, sets up socket communication, and processes client requests.

2. returnAnswer Method:

• Takes a mathematical expression as input and returns the computed result.

• Splits the input expression into operands and operator.

• Performs the corresponding arithmetic operation and returns the result.

• Handles division by zero by throwing an ArithmeticException.

Execution:

• To run the server, execute the main method of the Server class.

• The server listens for client connections on port 4008.

• Once a client connects, it receives mathematical expressions, computes the results, and sends them back to the client.

• The server terminates when the client sends "0/0=" to indicate the end of input or when an error occurs.

* A screen shot of a computer program

  Description automatically generatedClient Code Screenshots

A screen shot of a computer program

Description automatically generated

* Client Code description

Client Class Overview:

This Java class, Client, embodies a client-side application facilitating communication with a server. It establishes a connection to a server running on the localhost at port 4008, sends user input (mathematical expressions) to the server, and receives and displays responses.

Main Components:

1. Socket Setup:

• The client initializes a Socket object, connecting it to the server's address (localhost) and port 4008.

2. Input and Output Streams:

• It sets up input and output streams to interact with the server.

• The client reads input from the user via a BufferedReader reading from System.in.

• It also sets up a BufferedReader to read input from the server, and a PrintWriter to send output to the server.

3. Processing User Input:

• The client enters a loop where it continuously reads input from the user.

• If the user inputs "0/0=", indicating the end of input, the client sends this to the server and breaks out of the loop to end its execution.

• Otherwise, it sends the user's input to the server and awaits a response.

4. Displaying Responses:

• Upon receiving a response from the server, the client displays it to the user.

5. Error Handling:

• It catches and handles IOExceptions that might occur during socket communication, printing error messages if any exceptions arise.

Methods:

• Main Method (main):

• Serves as the entry point of the program.

• Initializes variables, sets up socket communication, and processes user input and server responses.

Execution:

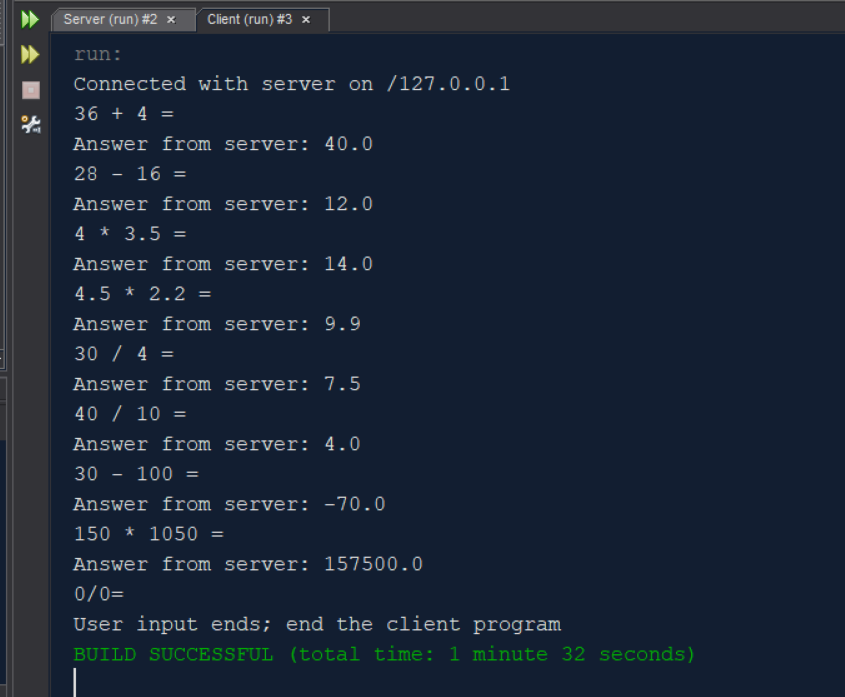
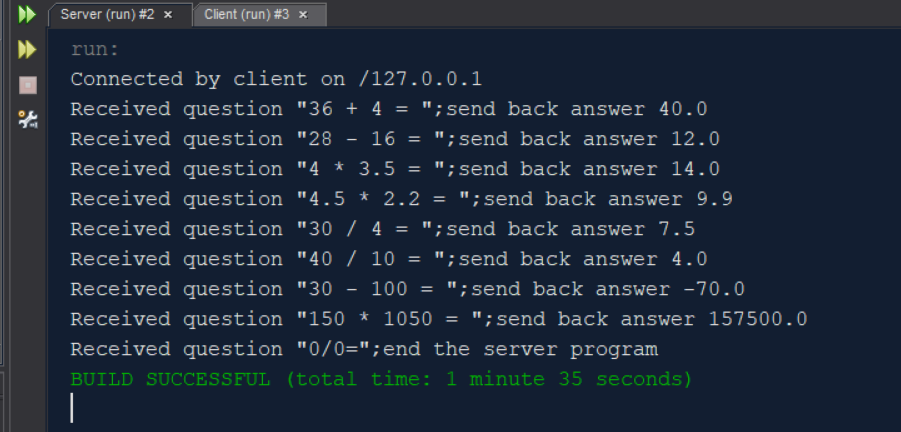
• To execute the client application, invoke the main method of the Client class.

• The client connects to the server running on localhost at port 4008.

• It prompts the user to input mathematical expressions.

• Upon receiving each response from the server, it displays it to the user.

• The client program terminates either when the user inputs "0/0=" to end the input or when an error occurs.

* Client Output
* Server Output
* **How to compile and run your code**

1. Compiling:

Server Class:

1. Open Apache NetBeans IDE.

2. Navigate to your project containing the Server.java file.

3. Right-click on the Server.java file in the "Projects" window.

4. Select "Build File" from the context menu.

Client Class:

1. Similarly, navigate to your project containing the Client.java file.

2. Right-click on the Client.java file in the "Projects" window.

3. Select "Build File" from the context menu.

2. Running:

Server Class:

1. After successful compilation, locate the Server class in the "Projects" window.

2. Right-click on the Server.java file.

3. Select "Run File" from the context menu.

Client Class:

1. After successful compilation, locate the Client class in the "Projects" window.

2. Right-click on the Client.java file.

3. Select "Run File" from the context menu.

3. Interacting:

Once both the server and client are running:

• Enter mathematical expressions in the client's console window.

• Press Enter after each expression.

• The server will compute the result and send it back to the client, displaying it in the client's console window.

• To end the client program, input "0/0=" and press Enter.

• To end the server program, simply close its console window or stop it from within NetBeans.

End

Thank You ^\_^