Socket Programming Lab Task

Objective: Implement a client-server application where the client sends arithmetic expressions to the server, and the server returns the computed result.

Task Requirements

Server Implementation

1. Create a TCP server that listens on **port 12345**.
2. Accept incoming client connections.
3. Receive arithmetic expressions in the format:
    - `"operand1 operator operand2"` (e.g., `"5 + 3"`, `"10 / 2"`).
4. Supported operations:
    - Addition (+), Subtraction (−), Multiplication (*), Division (/).
5. Handle errors gracefully:
    - Invalid expressions (e.g., `"5 +"`).
    - Division by zero.
6. Send the result back to the client.

Client Implementation

1. Connect to the server running on `localhost:12345`.
2. Prompt the user to input arithmetic expressions.
3. Send the expression to the server.
4. Display the server's response.
5. Allow the user to continue or quit by typing `"quit"`.

Example Workflow

1. **Server Output:**

plaintext

- `Server started on localhost:12345`

```
Connected by ('127.0.0.1', 54322)
```

- **Client Input/Output:**

plaintext

```
    2. Connected to server at localhost:12345
Enter expressions like '5 + 3' or 'quit' to exit:
> 5 + 3
Result: 8
> 10 / 0
Result: Error: Division by zero
> quit
```

Submission Guidelines

1. Submit two separate files:
    - `server.py` (Server implementation).
    - `client.py` (Client implementation).
2. Include comments explaining key steps.
3. Ensure the code runs without errors.

Bonus (Optional)

1. Extend the calculator to support modulus (`%`) and exponentiation (`^`).
2. Allow the server to handle multiple clients simultaneously (multi-threading).

Notes

- You can use any programming language
- Test your code thoroughly before submission.
- Document any assumptions or limitations.