**Aim**

The aim of this system is to demonstrate **Remote Procedure Call (RPC)** using the **XML-RPC protocol** in Python. The system functions as a **distributed calculator**, where the **Server Script** exposes basic arithmetic operations (add, subtract, multiply, divide) over a network interface, and the **Client Script** allows a user to remotely invoke these functions and receive the results.

**Short Algorithm (System)**

The system operates using two synchronized scripts:

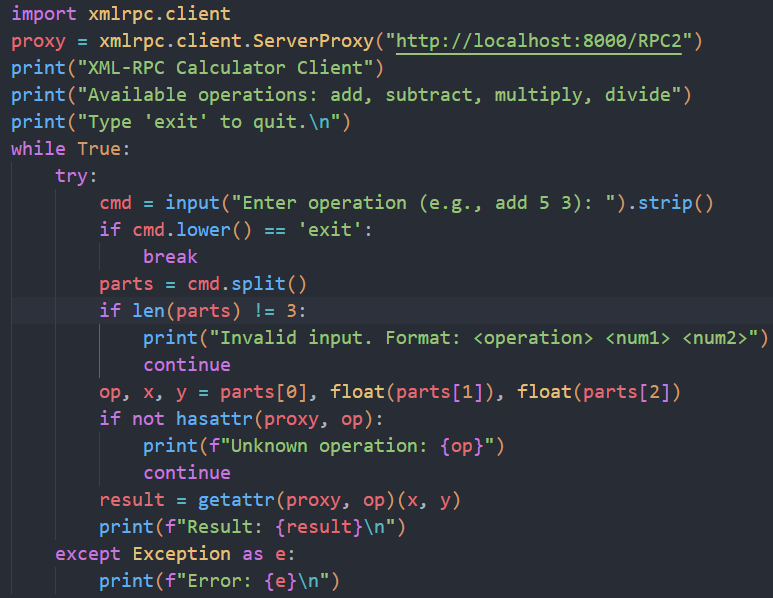
**Server Script Algorithm**

1. **Setup XML-RPC Server**: Import necessary XML-RPC classes and set the request path to /RPC2.
2. **Initialize Server**: Create a SimpleXMLRPCServer instance bound to localhost on port 8000.
3. **Define Functions**: Implement the calculator functions (add, subtract, multiply, divide), including error handling for division by zero.
4. **Register Functions**: Register the defined Python functions with the server, making them callable remotely by their string names (e.g., 'add').
5. **Start Service**: Call server.serve\_forever() to keep the server running and listening for client requests.

**Client Script Algorithm**

1. **Initialize Proxy**: Import xmlrpc.client and create a ServerProxy object connected to the server's URL (http://localhost:8000/RPC2).
2. **Start User Loop**: Enter an infinite loop to accept user input until the user types 'exit'.
3. **Process Input**:
   * Read the user command (e.g., add 5 3).
   * Parse the command into the **operation** (op) and two floating-point numbers (x, y).
4. **Validate and Call**:
   * Check if the **proxy object** has an attribute matching the requested op (i.e., if the function exists on the server).
   * If valid, dynamically call the remote function using getattr(proxy, op)(x, y).
5. **Display Result**: Print the returned result (or an error message if an exception occurs).

**Source Code(calculatorClient.py)**

****

**Source Code(calculatorServer.py)**

****

**Result (Sample Output)**

This output assumes the **Server Script** is running in one terminal and the **Client Script** is running in another.

