Supriya Patil

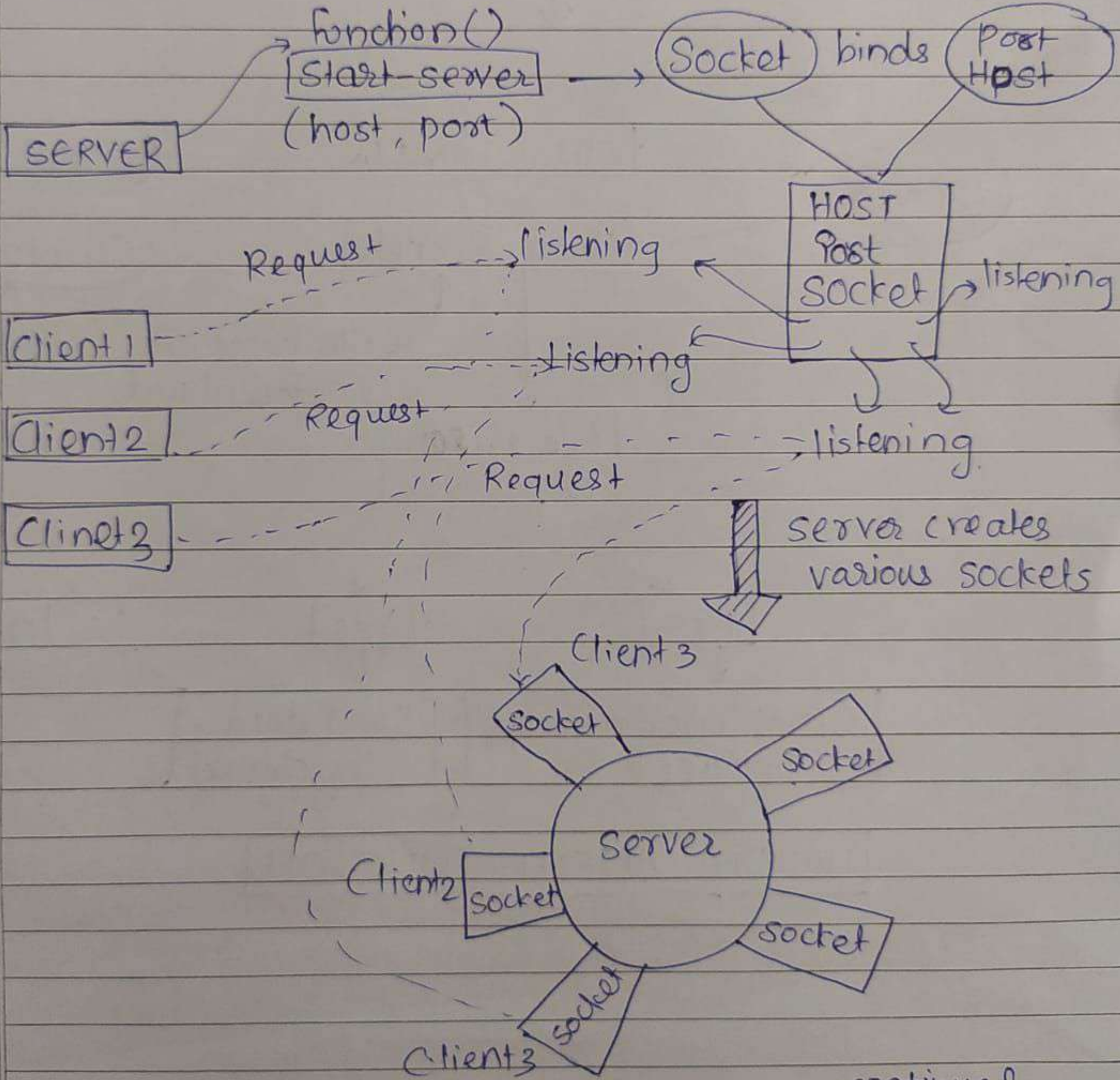
Page no.: _____/

Date: ___/___/_____/

JAVA AND PYTHON LAB 14

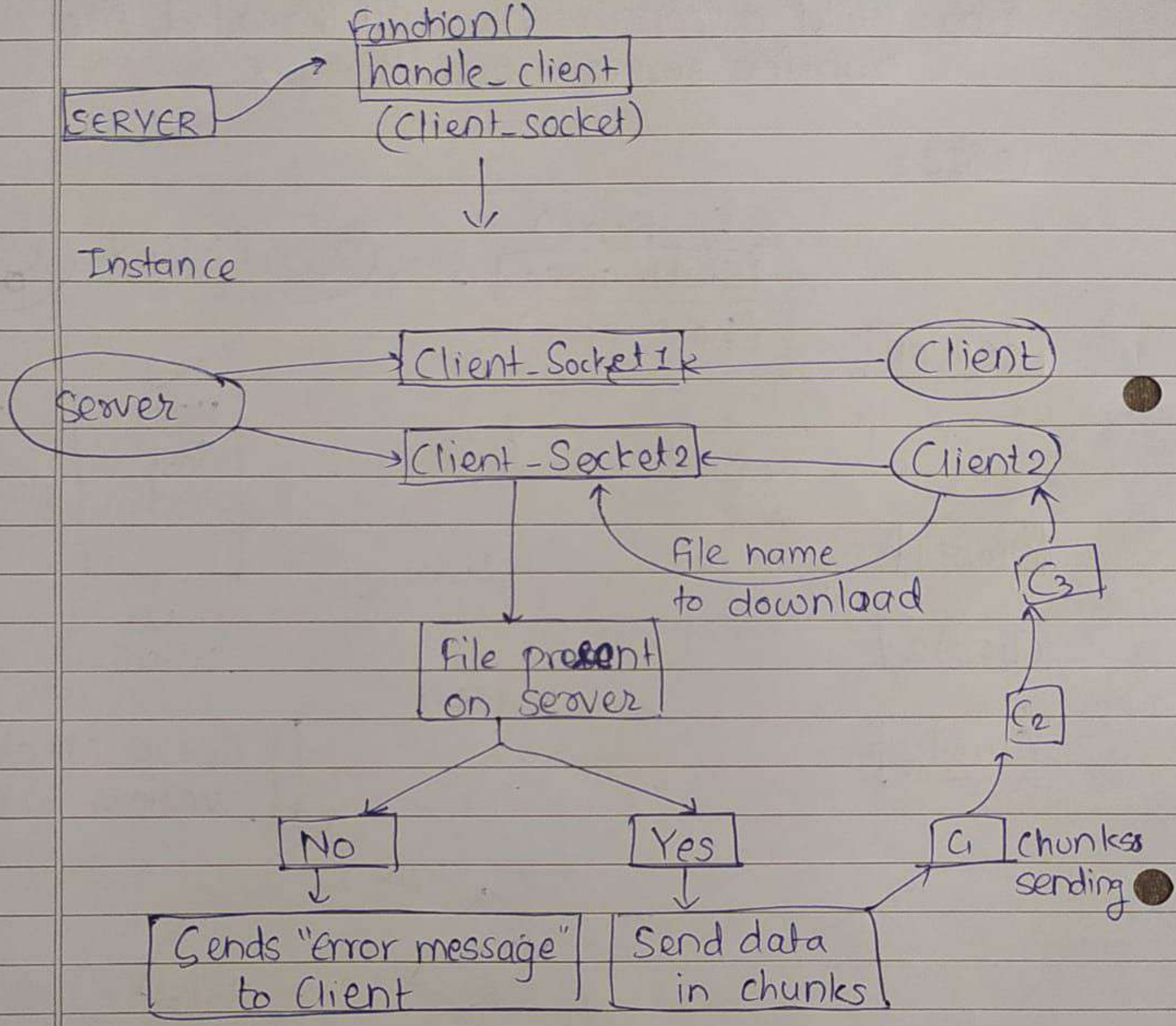
Aim: Build a concurrent multithreaded file transfer server using threads

LOGIC :



continued

~~SA~~ START-SERVER FUNCTION



HANDLE-CLIENT FUNCTION

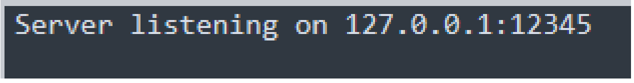
JAVA AND PYTHON LAB 14

Build a concurrent multithreaded file transfer server using threads

SOURCE CODE:

```
import socket
import threading
import os

def handle_client(client_socket):
    # Receive the file name from the client
    file_name = client_socket.recv(1024).decode()
    print(f"Received request for file: {file_name}")

    # Check if the file exists
    if os.path.isfile(file_name):
        # Open the file in binary mode
        with open(file_name, 'rb') as file:
            # Read the file data in chunks and send it to the client
            chunk = file.read(1024)
            while chunk:
                client_socket.send(chunk)
                chunk = file.read(1024)
    else:
        # File does not exist
        error_message = f"File '{file_name}' does not exist"
        client_socket.send(error_message.encode())

    # Close the client socket
    client_socket.close()

def start_server(host, port):
    # Create a socket object
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    # Bind the socket to a specific address and port
    server_socket.bind((host, port))

    # Listen for incoming connections
    server_socket.listen(5)
    print(f"Server listening on {host}:{port}")

    while True:
        # Accept a client connection
        client_socket, client_address = server_socket.accept()
        print(f"Accepted connection from {client_address[0]}:{client_address[1]}")

        # Start a new thread to handle the client
        client_thread = threading.Thread(target=handle_client, args=(client_socket,))
        client_thread.start()

if __name__ == '__main__':
    # Set the server host and port
    HOST = '127.0.0.1'  # localhost
    PORT = 12345

    # Start the server
    start_server(HOST, PORT)
```

OUTPUT



```
Server listening on 127.0.0.1:12345
```