5 Socket Programming - File Transfer (Multiple Clients)

Write a program to create a server that listens to port 5006 using stream sockets. Write a simple client program to connect to the server. Run multiple clients that request the server for binary files. The server should service each client one after the other before terminating the connection. Now, use a try-except clause and show that your program catches an exception for a file not found on the server.

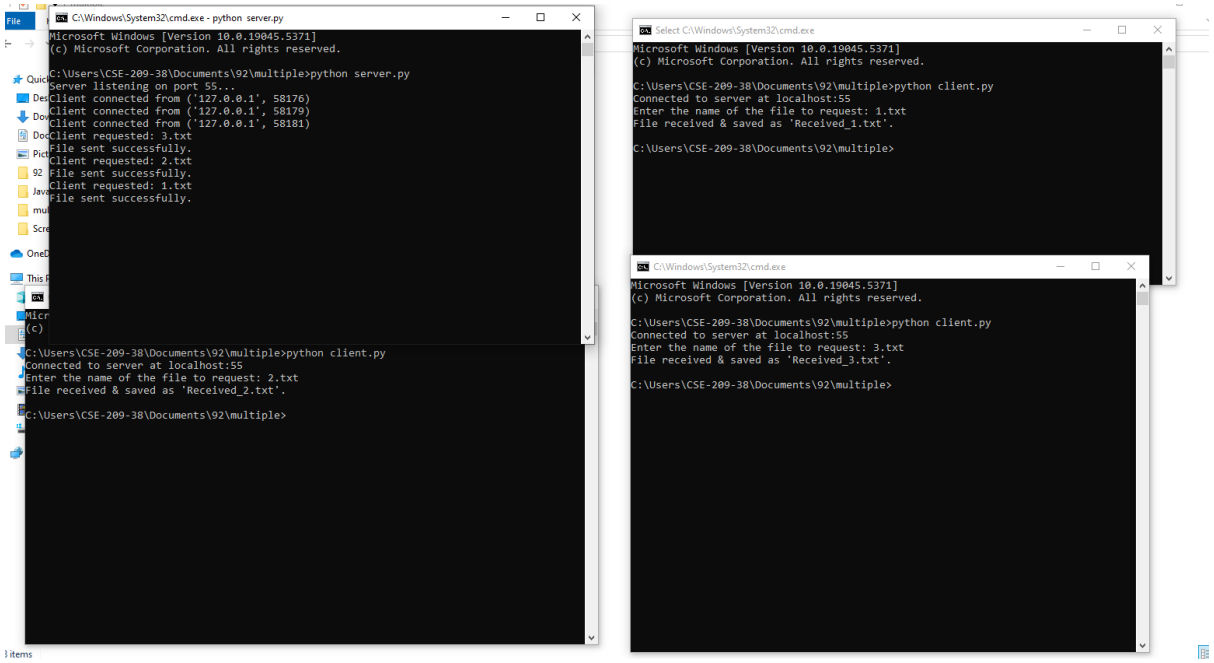
Server.py

```
import socket
import os
import threading
PORT = 55
def handle_client(client_socket, addr):
    print(f"Client connected from {addr}")
    file_name = client_socket.recv(1024).decode()
    print(f"Client requested: {file_name}")
    if os.path.exists(file_name):
        with open(file_name, "rb") as file:
            while (chunk := file.read(1024)):
                client_socket.sendall(chunk)
        print("File sent successfully.")
    else:
        print("File not found.")
        client_socket.sendall("File not found.".encode())

    client_socket.close()

def main():
    server_socket = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)
    server_socket.bind(("localhost", PORT))
    server_socket.listen(5)
```

```
    print(f"Server listening on port {PORT}...")


    try:
        while True:
            client_socket, addr = server_socket.accept()
            client_thread = threading.Thread(target=handle_client,
args=(client_socket, addr))
            client_thread.start()
    except KeyboardInterrupt:
        print("Server is shutting down.")
    finally:
        server_socket.close()


if __name__ == "__main__":
    main()
```

client.py

```
import socket


def main():
    server_address = "localhost"
    port = 55
    try:
        client_socket = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)
        client_socket.connect((server_address, port))
        print(f"Connected to server at {server_address}:{port}")
        file_name = input("Enter the name of the file to request: ")
        client_socket.sendall(file_name.encode())
        with open(f"Received_{file_name}", "wb") as file:
            while True:
                data = client_socket.recv(1024)
                if not data:
                    break  # Stop when no more data is received
                file.write(data)

        print(f"File received & saved as 'Received_{file_name}'.")

    except Exception as e:
        print("Error:", e)

    finally:
        client_socket.close()

if __name__ == "__main__":
    main()
```


C:\Windows\System32\cmd.exe - python server.py
Microsoft Windows [Version 10.0.19045.5371]
(c) Microsoft Corporation. All rights reserved.
C:\Users\CSE-209-38\Documents\92\multiple>python server.py
Server listening on port 55...
Client connected from ('127.0.0.1', 58176)
Client connected from ('127.0.0.1', 58179)
Client connected from ('127.0.0.1', 58181)
Client requested: 3.txt
File sent successfully.
Client requested: 2.txt
File sent successfully.
Client requested: 1.txt
File sent successfully.
C:\Users\CSE-209-38\Documents\92\multiple>python client.py
Connected to server at localhost:55
Enter the name of the file to request: 2.txt
File received & saved as 'Received_2.txt'.
C:\Users\CSE-209-38\Documents\92\multiple>
Select C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5371]
(c) Microsoft Corporation. All rights reserved.
C:\Users\CSE-209-38\Documents\92\multiple>python client.py
Connected to server at localhost:55
Enter the name of the file to request: 1.txt
File received & saved as 'Received_1.txt'.
C:\Users\CSE-209-38\Documents\92\multiple>
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5371]
(c) Microsoft Corporation. All rights reserved.
C:\Users\CSE-209-38\Documents\92\multiple>python client.py
Connected to server at localhost:55
Enter the name of the file to request: 3.txt
File received & saved as 'Received_3.txt'.
C:\Users\CSE-209-38\Documents\92\multiple>