# Files for TCP Socket Programs (Hello, File Transfer, Calculator)

---

## 1️⃣ hello_server.py

```
import socket

SERVER_IP = '127.0.0.12'
SERVER_PORT = 12000
BUFFER_SIZE = 1024

server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind((SERVER_IP, SERVER_PORT))
server_socket.listen(1)
print(f"Hello Server listening on {SERVER_IP}:{SERVER_PORT}...")

conn, addr = server_socket.accept()
print(f"Connected to: {addr}")

# Exchange hello
conn.send(b"Hello from server!")
client_msg = conn.recv(BUFFER_SIZE).decode()
print(f"Client says: {client_msg}")

conn.close()
server_socket.close()
```

## 2️⃣ hello_client.py

```
import socket

SERVER_IP = '127.0.0.12'
SERVER_PORT = 12000
BUFFER_SIZE = 1024

client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client_socket.connect((SERVER_IP, SERVER_PORT))

server_msg = client_socket.recv(BUFFER_SIZE).decode()
print(f"Server: {server_msg}")

client_msg = input("Say hello back: ")
client_socket.send(client_msg.encode())
```

```
client_socket.close()
```

## 3️⃣ file_server.py

```
import socket

SERVER_IP = '127.0.0.12'
SERVER_PORT = 13000
BUFFER_SIZE = 1024

server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind((SERVER_IP, SERVER_PORT))
server_socket.listen(1)
print(f"File Server listening on {SERVER_IP}:{SERVER_PORT}...")

conn, addr = server_socket.accept()
print(f"Connected to: {addr}")

conn.send(b"Send filename: ")
filename = conn.recv(BUFFER_SIZE).decode()
conn.send(b"Ready to receive file...")

with open(f"server_{filename}", 'wb') as f:
    while True:
        data = conn.recv(BUFFER_SIZE)
        if not data:
            break
        f.write(data)

print(f"Received file: server_{filename}")
conn.send(b"File received successfully.")
conn.close()
server_socket.close()
```

## 4️⃣ file_client.py

```
import socket
import os

SERVER_IP = '127.0.0.12'
SERVER_PORT = 13000
BUFFER_SIZE = 1024

client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client_socket.connect((SERVER_IP, SERVER_PORT))

prompt = client_socket.recv(BUFFER_SIZE).decode()
```

```
filename = input(prompt + " ")
client_socket.send(filename.encode())

ready_msg = client_socket.recv(BUFFER_SIZE).decode()
print(ready_msg)

with open(filename, 'rb') as f:
    data = f.read(BUFFER_SIZE)
    while data:
        client_socket.send(data)
        data = f.read(BUFFER_SIZE)

client_socket.shutdown(socket.SHUT_WR)
print(client_socket.recv(BUFFER_SIZE).decode())
client_socket.close()
```

## 5️⃣ calc_server.py

```
import socket
import math

SERVER_IP = '127.0.0.12'
SERVER_PORT = 14000
BUFFER_SIZE = 1024

server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind((SERVER_IP, SERVER_PORT))
server_socket.listen(1)
print(f"Calculator Server listening on {SERVER_IP}:{SERVER_PORT}...")

conn, addr = server_socket.accept()
print(f"Connected to: {addr}")

# Operation type
conn.send(b"Choose operation type (arithmetic/trigonometric): ")
op_type = conn.recv(BUFFER_SIZE).decode().strip().lower()

if op_type == 'arithmetic':
    conn.send(b"Enter expression (e.g., 5 + 3): ")
    expr = conn.recv(BUFFER_SIZE).decode()
    try:
        result = eval(expr)
        conn.send(f"Result: {result}".encode())
    except Exception as e:
        conn.send(f"Error: {e}".encode())

elif op_type == 'trigonometric':
    conn.send(b"Enter function and value (e.g., sin 30): ")
    func_val = conn.recv(BUFFER_SIZE).decode().split()
    if len(func_val) != 2:
```

```
        conn.send(b"Invalid input")
    else:
        func, val = func_val[0].lower(), float(func_val[1])
        val_rad = math.radians(val)
        try:
            if func == 'sin':
                result = math.sin(val_rad)
            elif func == 'cos':
                result = math.cos(val_rad)
            elif func == 'tan':
                result = math.tan(val_rad)
            else:
                conn.send(b"Unsupported function")
                conn.close()
                exit()
            conn.send(f"Result: {result}".encode())
        except Exception as e:
            conn.send(f"Error: {e}".encode())

conn.close()
server_socket.close()
```

## 6️⃣ calc_client.py

```
import socket

SERVER_IP = '127.0.0.12'
SERVER_PORT = 14000
BUFFER_SIZE = 1024

client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client_socket.connect((SERVER_IP, SERVER_PORT))

prompt = client_socket.recv(BUFFER_SIZE).decode()
op_type = input(prompt + " ")
client_socket.send(op_type.encode())

prompt2 = client_socket.recv(BUFFER_SIZE).decode()
expr = input(prompt2 + " ")
client_socket.send(expr.encode())

result = client_socket.recv(BUFFER_SIZE).decode()
print(result)

client_socket.close()
```