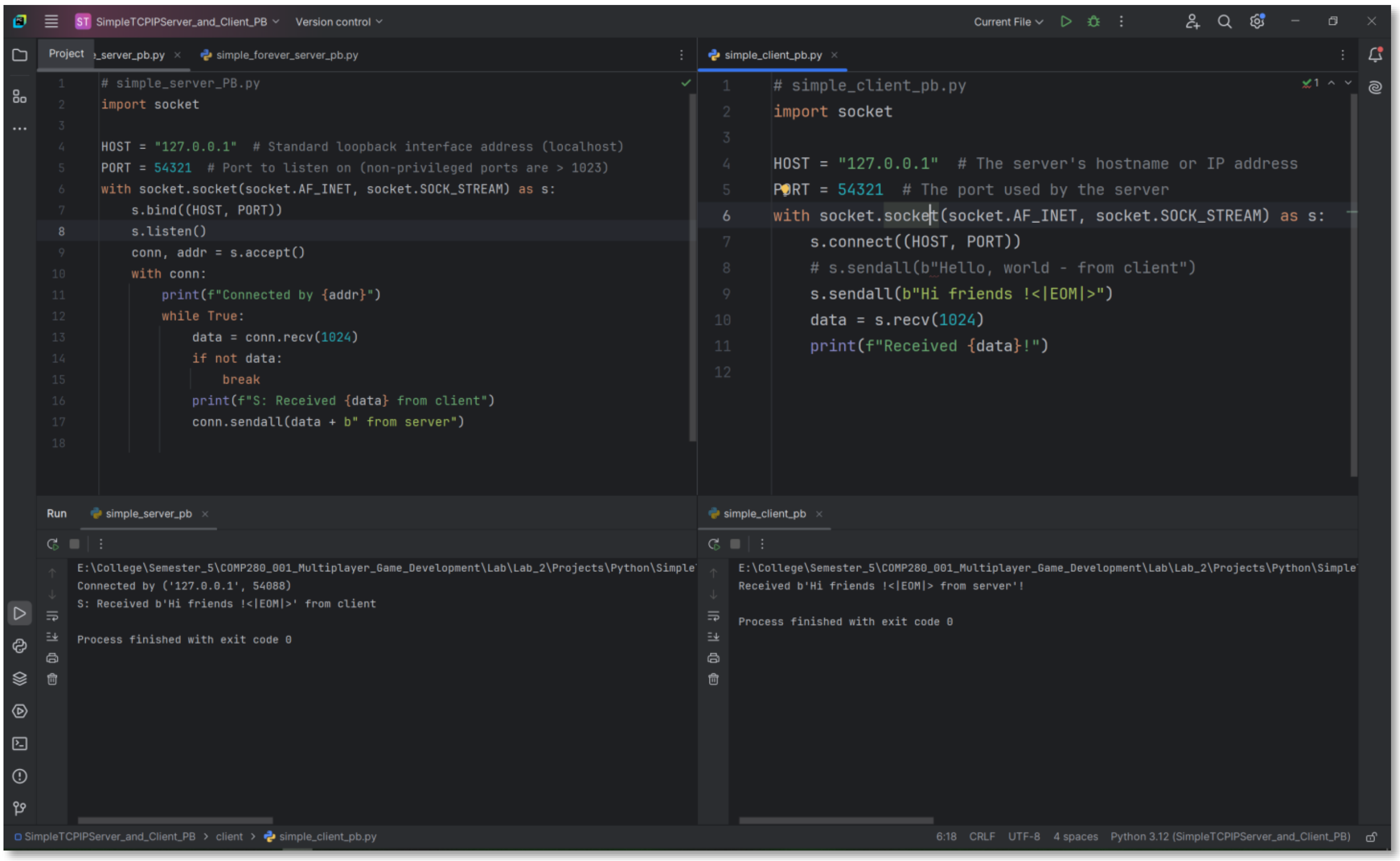
LAB – 2

SOCKETS

Implementation of a Simple TCPIP Server and TCPIP Client.

1.Python



```python
# simple_server_PB.py
import socket

HOST = "127.0.0.1"  # Standard loopback interface address (localhost)
PORT = 54321  # Port to listen on (non-privileged ports are > 1023)
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((HOST, PORT))
    s.listen()
    conn, addr = s.accept()
    with conn:
        print(f"Connected by {addr}")
        while True:
            data = conn.recv(1024)
            if not data:
                break
            print(f"S: Received {data} from client")
            conn.sendall(data + b" from server")
```

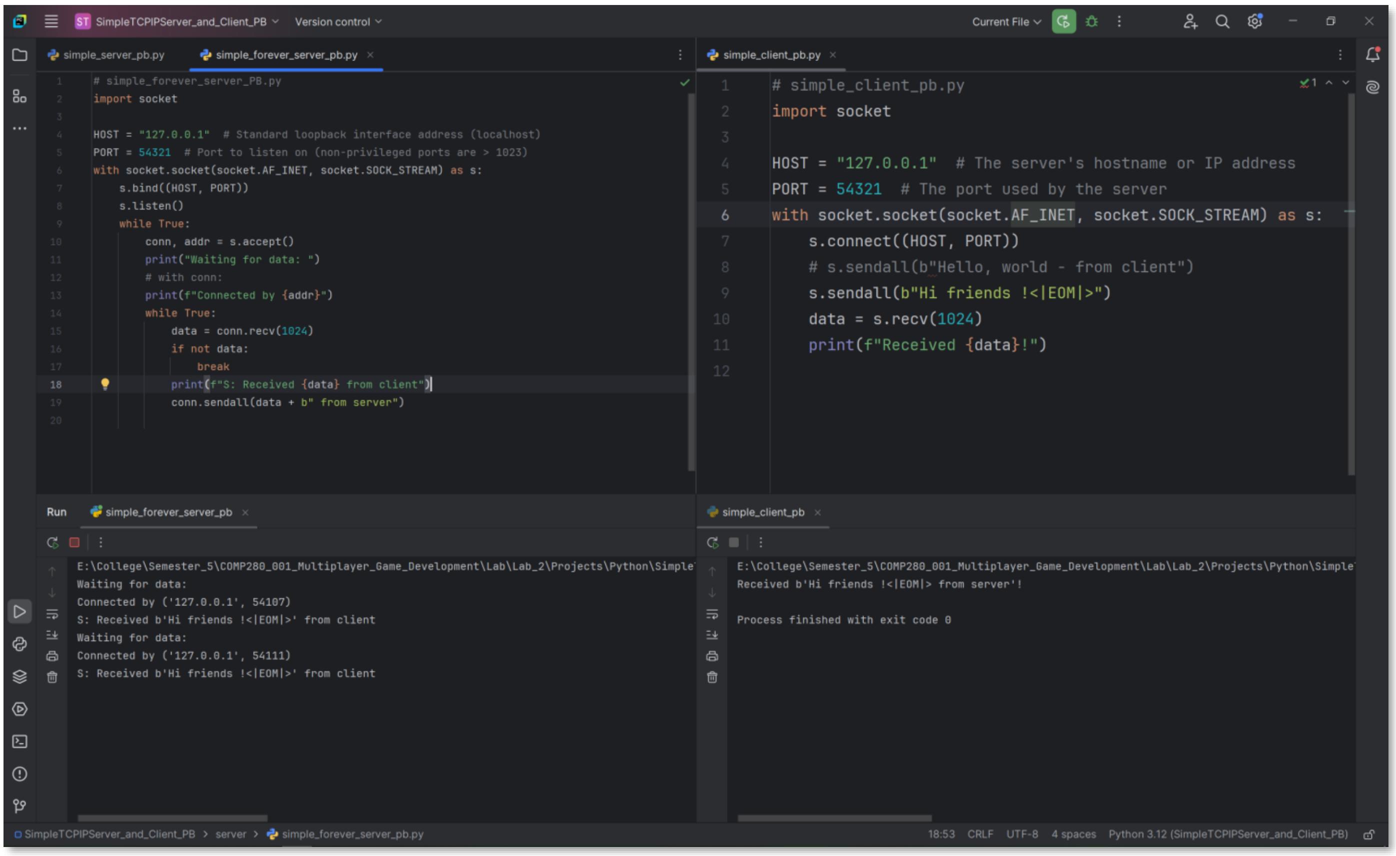
```python
# simple_client_pb.py
import socket

HOST = "127.0.0.1"  # The server's hostname or IP address
PORT = 54321  # The port used by the server
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((HOST, PORT))
    # s.sendall(b"Hello, world - from client")
    s.sendall(b"Hi friends !<|EOM|>")
    data = s.recv(1024)
    print(f"Received {data}!")
```

```
E:\College\Semester_5\COMP280_001_Multiplayer_Game_Development\Lab\Lab_2\Projects\Python\Simple
Connected by ('127.0.0.1', 54088)
S: Received b'Hi friends !<|EOM|>' from client

Process finished with exit code 0
```

```
E:\College\Semester_5\COMP280_001_Multiplayer_Game_Development\Lab\Lab_2\Projects\Python\Simple
Received b'Hi friends !<|EOM|> from server'!

Process finished with exit code 0
```

```
SimpleTCPIPServer_and_Client_PB > client > simple_client_pb.py
6:18  CRLF  UTF-8  4 spaces  Python 3.12 (SimpleTCPIPServer_and_Client_PB)
```

Fig 1.1: Python socket communication between Simple Client to Simple Server.



```python
# simple_forever_server_PB.py
import socket

HOST = "127.0.0.1"  # Standard loopback interface address (localhost)
PORT = 54321  # Port to listen on (non-privileged ports are > 1023)
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((HOST, PORT))
    s.listen()
    while True:
        conn, addr = s.accept()
        print("Waiting for data: ")
        # with conn:
        print(f"Connected by {addr}")
        while True:
            data = conn.recv(1024)
            if not data:
                break
            print(f"S: Received {data} from client")
            conn.sendall(data + b" from server")
```

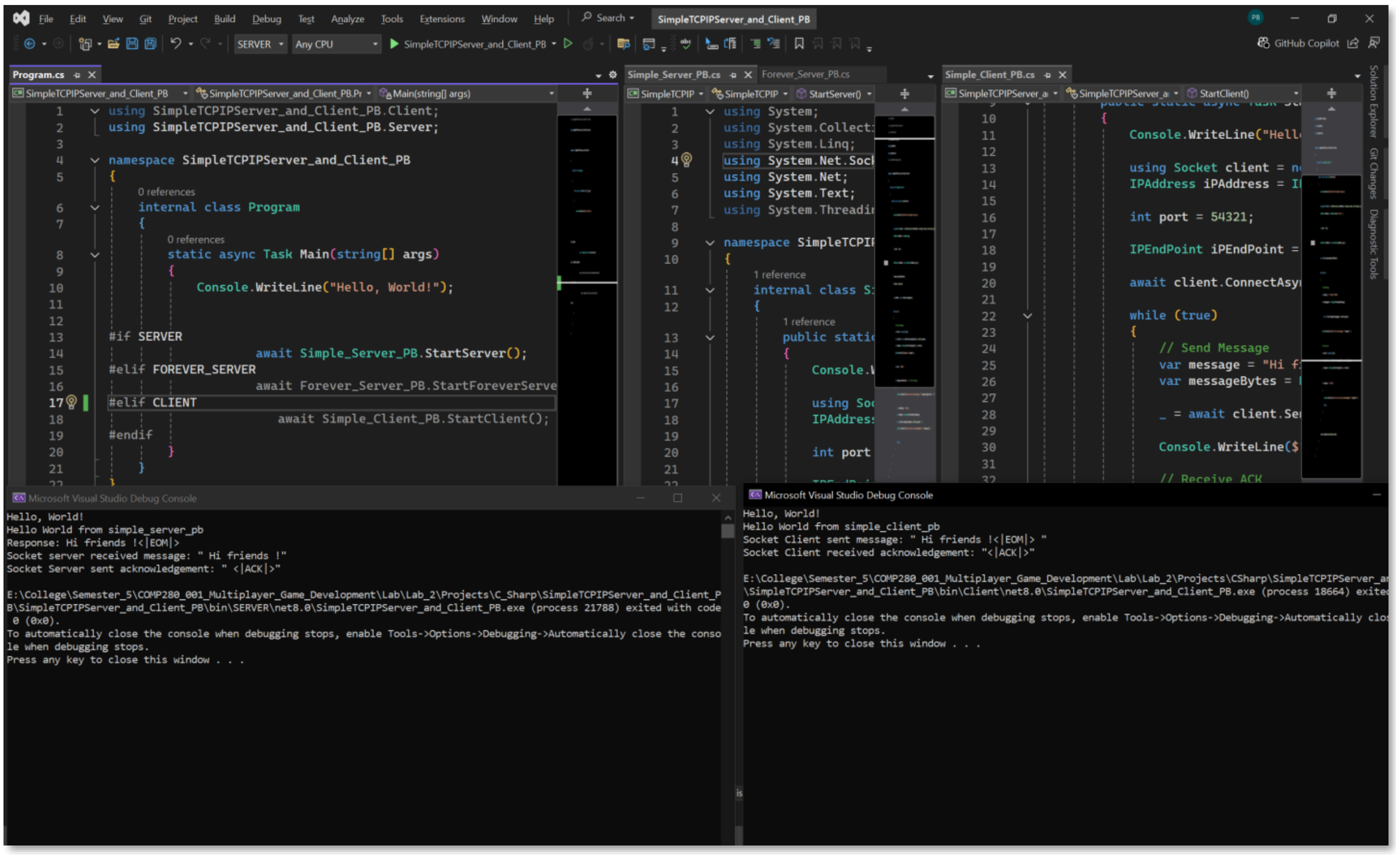
```python
# simple_client_pb.py
import socket

HOST = "127.0.0.1"  # The server's hostname or IP address
PORT = 54321  # The port used by the server
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((HOST, PORT))
    # s.sendall(b"Hello, world - from client")
    s.sendall(b"Hi friends !<|EOM|>")
    data = s.recv(1024)
    print(f"Received {data}!")
```

```
E:\College\Semester_5\COMP280_001_Multiplayer_Game_Development\Lab\Lab_2\Projects\Python\Simple
Waiting for data:
Connected by ('127.0.0.1', 54107)
S: Received b'Hi friends !<|EOM|>' from client
Waiting for data:
Connected by ('127.0.0.1', 54111)
S: Received b'Hi friends !<|EOM|>' from client
```

```
E:\College\Semester_5\COMP280_001_Multiplayer_Game_Development\Lab\Lab_2\Projects\Python\Simple
Received b'Hi friends !<|EOM|> from server'!

Process finished with exit code 0
```

Fig 1.2: Python socket communication between Simple Client to Simple ***Forever*** *Server.*

2.C#



```csharp
using SimpleTCPIPServer_and_Client_PB.Client;
using SimpleTCPIPServer_and_Client_PB.Server;

namespace SimpleTCPIPServer_and_Client_PB
{
    internal class Program
    {
        static async Task Main(string[] args)
        {
            Console.WriteLine("Hello, World!");


#if SERVER
                    await Simple_Server_PB.StartServer();
#elif FOREVER_SERVER
                    await Forever_Server_PB.StartForeverServe
#elif CLIENT
                        await Simple_Client_PB.StartClient();
#endif
        }
    }
```

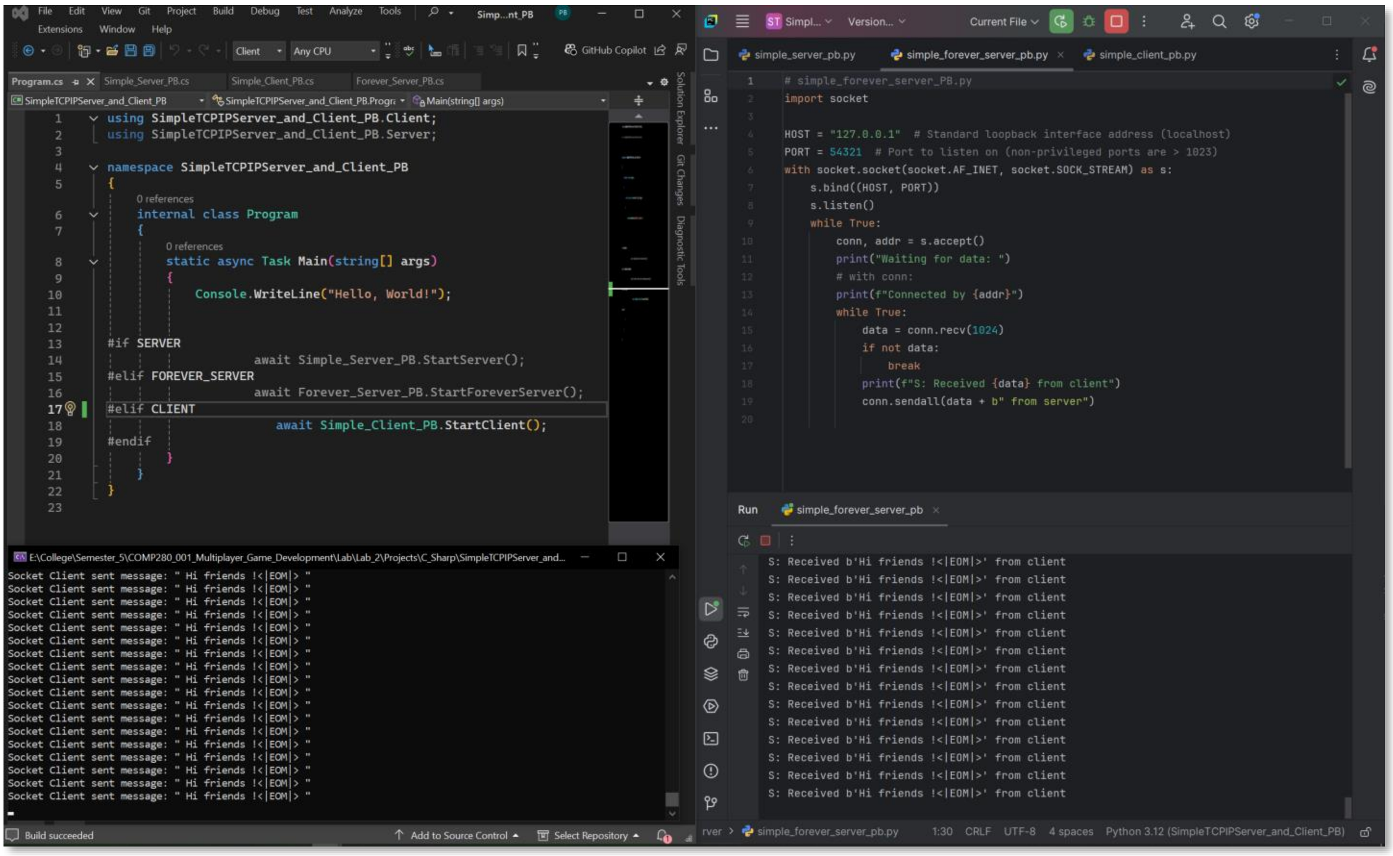
```csharp
using System;
using System.Collect
using System.Linq;
using System.Net.Sock
using System.Net;
using System.Text;
using System.Threadir

namespace SimpleTCPIF
{
    internal class S
    {
        public stati
        {
            Console.W

            using Soc
            IPAddress

            int port
```

```csharp
        {
            Console.WriteLine("Hell

            using Socket client = n
            IPAddress iPAddress = I

            int port = 54321;

            IPEndPoint iPEndPoint =

            await client.ConnectAsy

            while (true)
            {
                // Send Message
                var message = "Hi f
                var messageBytes = 

                _ = await client.Se

                Console.WriteLine($

                // Receive ACK
```

```
Hello, World!
Hello World from simple_server_pb
Response: Hi friends !<|EOM|>
Socket server received message: " Hi friends !"
Socket Server sent acknowledgement: " <|ACK|>"

E:\College\Semester_5\COMP280_001_Multiplayer_Game_Development\Lab\Lab_2\Projects\C_Sharp\SimpleTCPIPServer_and_Client_P
B\SimpleTCPIPServer_and_Client_PB\bin\SERVER\net8.0\SimpleTCPIPServer_and_Client_PB.exe (process 21788) exited with code
 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

```
Hello, World!
Hello World from simple_client_pb
Socket Client sent message: " Hi friends !<|EOM|> "
Socket Client received acknowledgement: "<|ACK|>"

E:\College\Semester_5\COMP280_001_Multiplayer_Game_Development\Lab\Lab_2\Projects\CSharp\SimpleTCPIPServer_an
\SimpleTCPIPServer_and_Client_PB\bin\Client\net8.0\SimpleTCPIPServer_and_Client_PB.exe (process 18664) exited
0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically clos
le when debugging stops.
Press any key to close this window . . .
```

Fig 2.1: C# Server and Client.



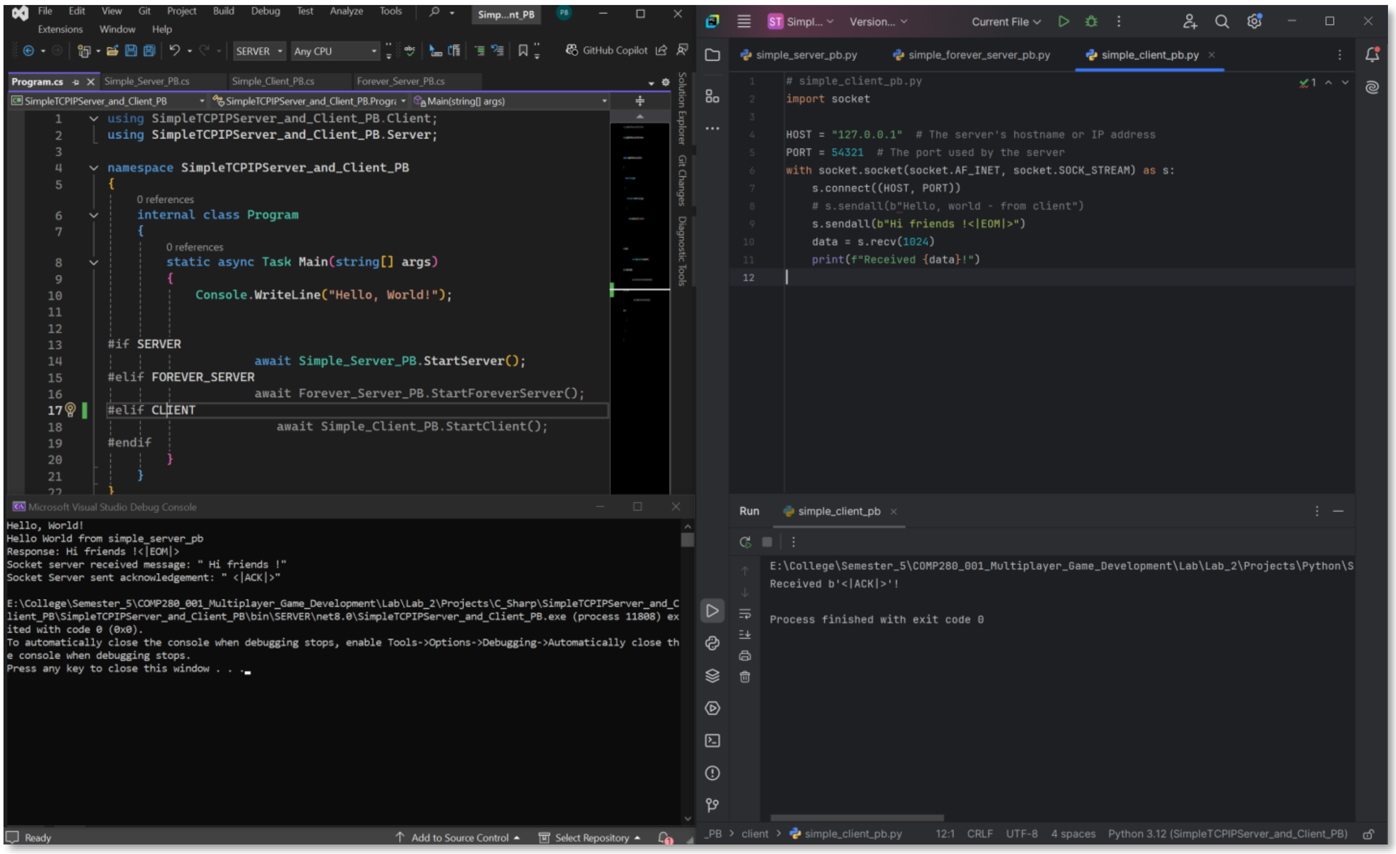
```csharp
using SimpleTCPIPServer_and_Client_PB.Client;
using SimpleTCPIPServer_and_Client_PB.Server;

namespace SimpleTCPIPServer_and_Client_PB
{
    internal class Program
    {
        static async Task Main(string[] args)
        {
            Console.WriteLine("Hello, World!");


#if SERVER
                    await Simple_Server_PB.StartServer();
#elif FOREVER_SERVER
                    await Forever_Server_PB.StartForeverServer();
#elif CLIENT
                       await Simple_Client_PB.StartClient();
#endif
        }
    }
}
```

```
Socket Client sent message: " Hi friends !<|EOM|> "
Socket Client sent message: " Hi friends !<|EOM|> "
Socket Client sent message: " Hi friends !<|EOM|> "
Socket Client sent message: " Hi friends !<|EOM|> "
Socket Client sent message: " Hi friends !<|EOM|> "
Socket Client sent message: " Hi friends !<|EOM|> "
Socket Client sent message: " Hi friends !<|EOM|> "
Socket Client sent message: " Hi friends !<|EOM|> "
Socket Client sent message: " Hi friends !<|EOM|> "
Socket Client sent message: " Hi friends !<|EOM|> "
Socket Client sent message: " Hi friends !<|EOM|> "
Socket Client sent message: " Hi friends !<|EOM|> "
Socket Client sent message: " Hi friends !<|EOM|> "
Socket Client sent message: " Hi friends !<|EOM|> "
Socket Client sent message: " Hi friends !<|EOM|> "
Socket Client sent message: " Hi friends !<|EOM|> "
Socket Client sent message: " Hi friends !<|EOM|> "
Socket Client sent message: " Hi friends !<|EOM|> "
```

```python
# simple_forever_server_PB.py
import socket

HOST = "127.0.0.1"  # Standard loopback interface address (localhost)
PORT = 54321  # Port to listen on (non-privileged ports are > 1023)
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((HOST, PORT))
    s.listen()
    while True:
        conn, addr = s.accept()
        print("Waiting for data: ")
        # with conn:
        print(f"Connected by {addr}")
        while True:
            data = conn.recv(1024)
            if not data:
                break
            print(f"S: Received {data} from client")
            conn.sendall(data + b" from server")
```

```
S: Received b'Hi friends !<|EOM|>' from client
S: Received b'Hi friends !<|EOM|>' from client
S: Received b'Hi friends !<|EOM|>' from client
S: Received b'Hi friends !<|EOM|>' from client
S: Received b'Hi friends !<|EOM|>' from client
S: Received b'Hi friends !<|EOM|>' from client
S: Received b'Hi friends !<|EOM|>' from client
S: Received b'Hi friends !<|EOM|>' from client
S: Received b'Hi friends !<|EOM|>' from client
S: Received b'Hi friends !<|EOM|>' from client
S: Received b'Hi friends !<|EOM|>' from client
S: Received b'Hi friends !<|EOM|>' from client
S: Received b'Hi friends !<|EOM|>' from client
S: Received b'Hi friends !<|EOM|>' from client
```

Fig 2.2: Python Server and C# Client.



```csharp
using SimpleTCPIPServer_and_Client_PB.Client;
using SimpleTCPIPServer_and_Client_PB.Server;

namespace SimpleTCPIPServer_and_Client_PB
{
    internal class Program
    {
        static async Task Main(string[] args)
        {
            Console.WriteLine("Hello, World!");


#if SERVER
                    await Simple_Server_PB.StartServer();
#elif FOREVER_SERVER
                    await Forever_Server_PB.StartForeverServer();
#elif CLIENT
                       await Simple_Client_PB.StartClient();
#endif
        }
    }
```

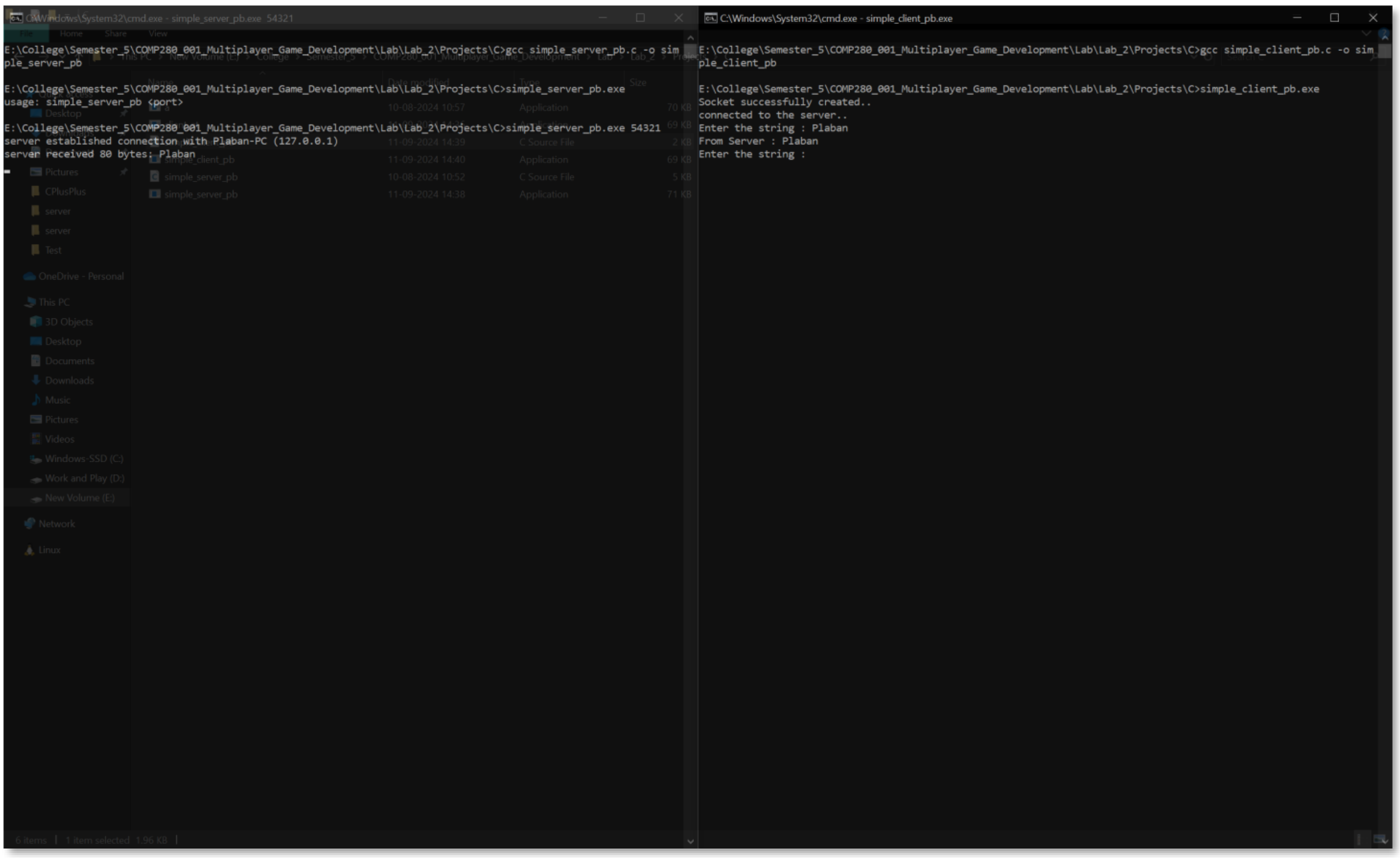
```
Hello, World!
Hello World from simple_server_pb
Response: Hi friends !<|EOM|>
Socket server received message: " Hi friends !"
Socket Server sent acknowledgement: " <|ACK|>"

E:\College\Semester_5\COMP280_001_Multiplayer_Game_Development\Lab\Lab_2\Projects\C_Sharp\SimpleTCPIPServer_and_C
lient_PB\SimpleTCPIPServer_and_Client_PB\bin\SERVER\net8.0\SimpleTCPIPServer_and_Client_PB.exe (process 11808) ex
ited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close th
e console when debugging stops.
Press any key to close this window . . .
```

```python
# simple_client_pb.py
import socket

HOST = "127.0.0.1"  # The server's hostname or IP address
PORT = 54321  # The port used by the server
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((HOST, PORT))
    # s.sendall(b"Hello, world - from client")
    s.sendall(b"Hi friends !<|EOM|>")
    data = s.recv(1024)
    print(f"Received {data}!")
```

```
E:\College\Semester_5\COMP280_001_Multiplayer_Game_Development\Lab\Lab_2\Projects\Python\S
Received b'<|ACK|>'!

Process finished with exit code 0
```

Fig 2.3: C# Server and Python Client.

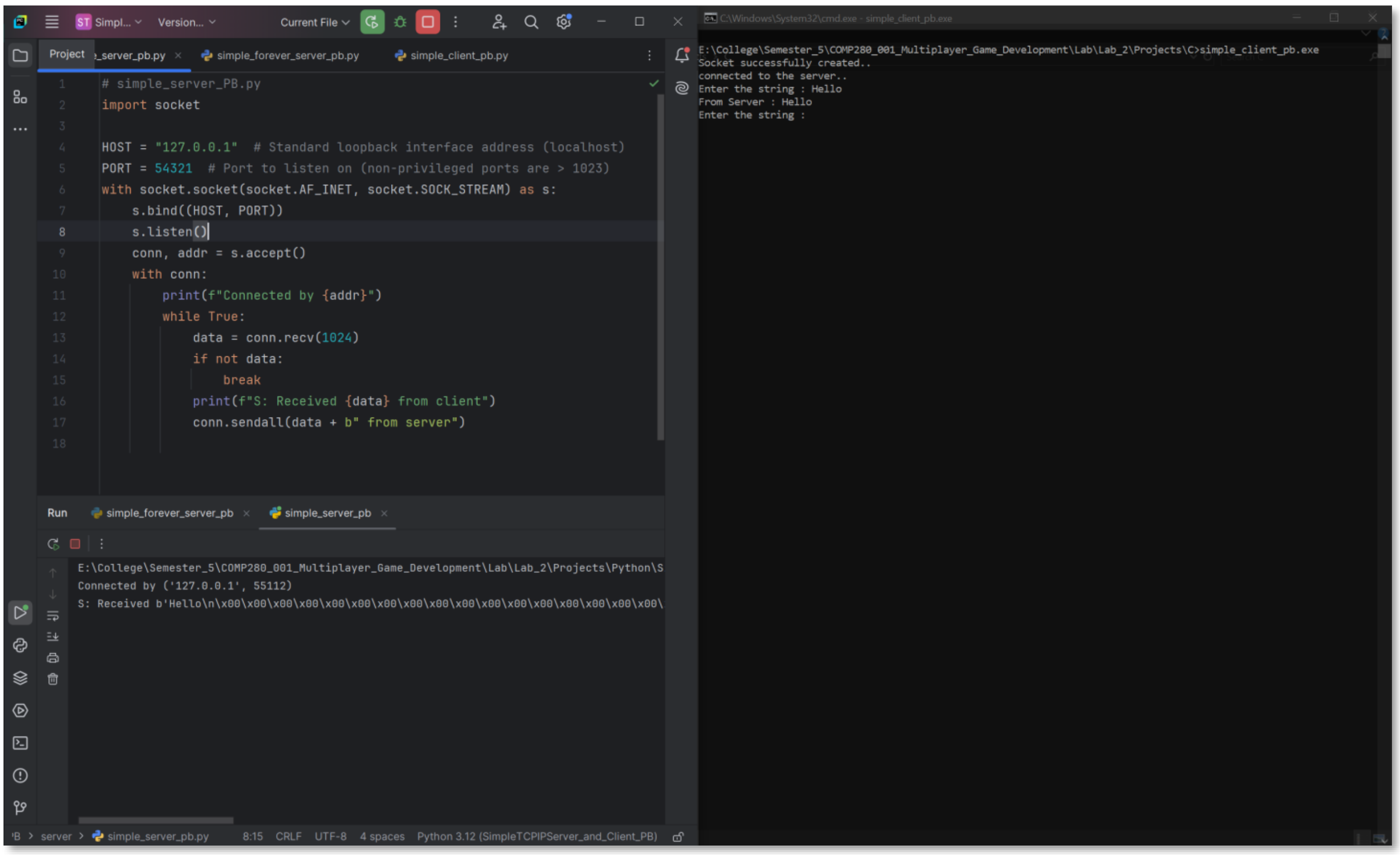
3. C



```
E:\College\Semester_5\COMP280_001_Multiplayer_Game_Development\Lab\Lab_2\Projects\C>gcc simple_server_pb.c -o sim
ple_server_pb

E:\College\Semester_5\COMP280_001_Multiplayer_Game_Development\Lab\Lab_2\Projects\C>simple_server_pb.exe
usage: simple_server_pb <port>

E:\College\Semester_5\COMP280_001_Multiplayer_Game_Development\Lab\Lab_2\Projects\C>simple_server_pb.exe 54321
server established connection with Plaban-PC (127.0.0.1)
server received 80 bytes: Plaban
```

```
E:\College\Semester_5\COMP280_001_Multiplayer_Game_Development\Lab\Lab_2\Projects\C>gcc simple_client_pb.c -o sim
ple_client_pb

E:\College\Semester_5\COMP280_001_Multiplayer_Game_Development\Lab\Lab_2\Projects\C>simple_client_pb.exe
Socket successfully created..
connected to the server..
Enter the string : Plaban
From Server : Plaban
Enter the string :
```

Fig 3.1: C Server and Client.

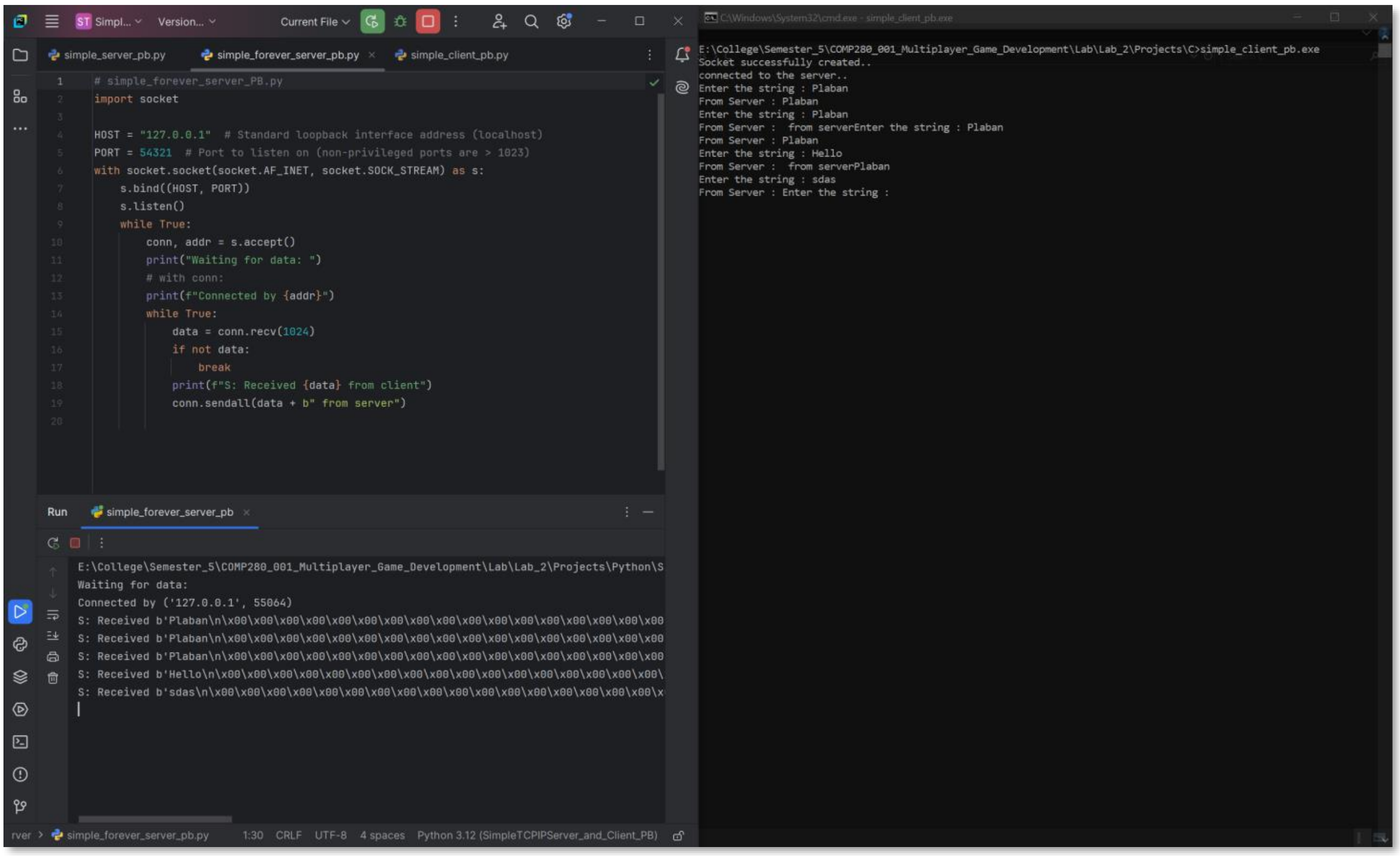


```python
# simple_server_PB.py
import socket

HOST = "127.0.0.1"  # Standard loopback interface address (localhost)
PORT = 54321  # Port to listen on (non-privileged ports are > 1023)
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((HOST, PORT))
    s.listen()
    conn, addr = s.accept()
    with conn:
        print(f"Connected by {addr}")
        while True:
            data = conn.recv(1024)
            if not data:
                break
            print(f"S: Received {data} from client")
            conn.sendall(data + b" from server")
```

```
E:\College\Semester_5\COMP280_001_Multiplayer_Game_Development\Lab\Lab_2\Projects\Python\S
Connected by ('127.0.0.1', 55112)
S: Received b'Hello\n\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
```

```
E:\College\Semester_5\COMP280_001_Multiplayer_Game_Development\Lab\Lab_2\Projects\C>simple_client_pb.exe
Socket successfully created..
connected to the server..
Enter the string : Hello
From Server : Hello
Enter the string :
```

Fig 3.2: Python Simple Server and C Client.

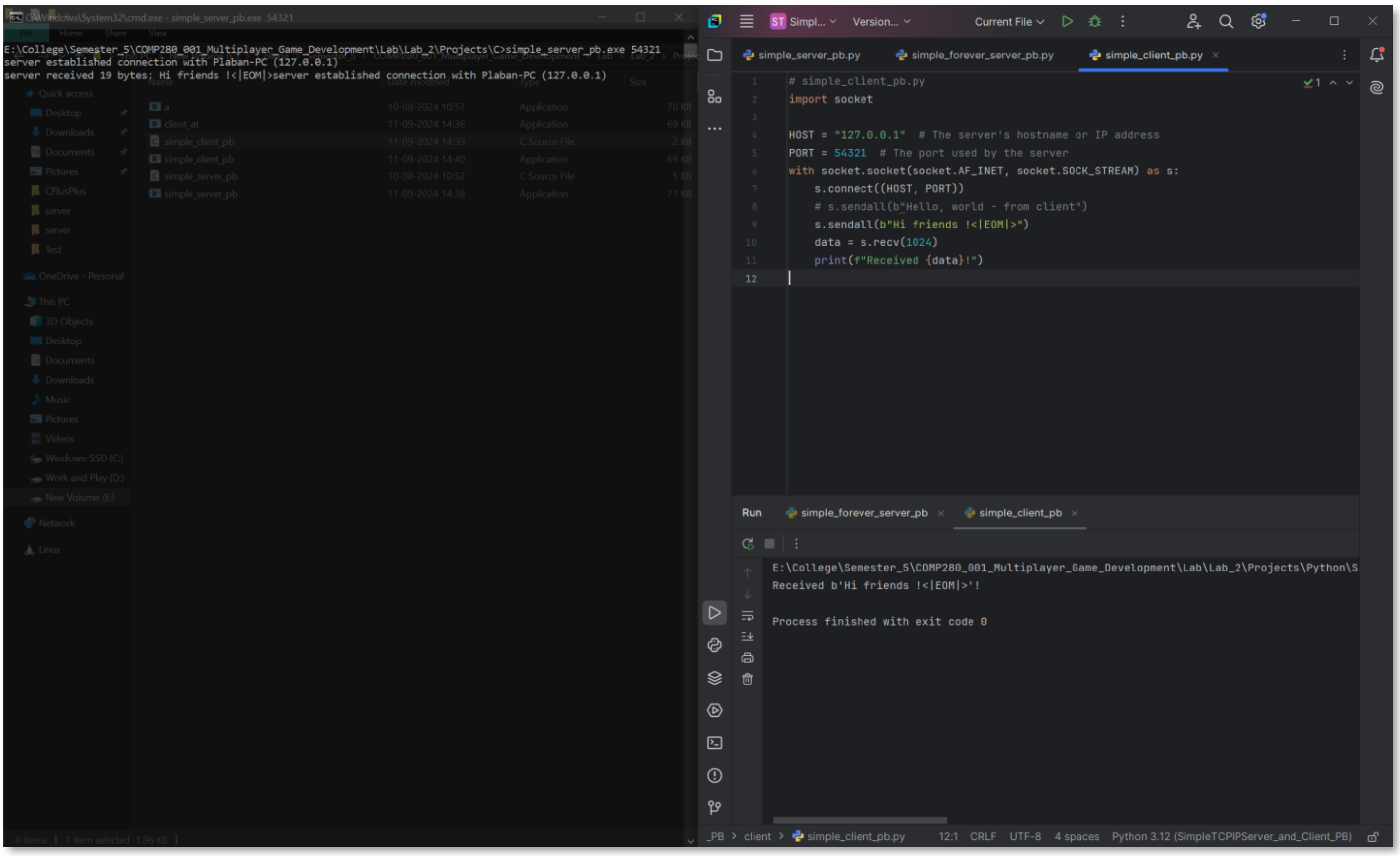

Simpl...
Version...
Current File
simple_server_pb.py
simple_forever_server_pb.py
simple_client_pb.py
Run
simple_forever_server_pb
C:\Windows\System32\cmd.exe - simple_client_pb.exe
rver > simple_forever_server_pb.py 1:30 CRLF UTF-8 4 spaces Python 3.12 (SimpleTCPIPServer_and_Client_PB)

```python
# simple_forever_server_PB.py
import socket

HOST = "127.0.0.1"  # Standard loopback interface address (localhost)
PORT = 54321  # Port to listen on (non-privileged ports are > 1023)
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((HOST, PORT))
    s.listen()
    while True:
        conn, addr = s.accept()
        print("Waiting for data: ")
        # with conn:
        print(f"Connected by {addr}")
        while True:
            data = conn.recv(1024)
            if not data:
                break
            print(f"S: Received {data} from client")
            conn.sendall(data + b" from server")
```

```
E:\College\Semester_5\COMP280_001_Multiplayer_Game_Development\Lab\Lab_2\Projects\Python\S
Waiting for data:
Connected by ('127.0.0.1', 55064)
S: Received b'Plaban\n\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
S: Received b'Plaban\n\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
S: Received b'Plaban\n\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
S: Received b'Hello\n\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
S: Received b'sdas\n\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
```

```
E:\College\Semester_5\COMP280_001_Multiplayer_Game_Development\Lab\Lab_2\Projects\C>simple_client_pb.exe
Socket successfully created..
connected to the server..
Enter the string : Plaban
From Server : Plaban
Enter the string : Plaban
From Server :  from serverEnter the string : Plaban
From Server : Plaban
Enter the string : Hello
From Server :  from serverPlaban
Enter the string : sdas
From Server : Enter the string :
```

Fig 3.3: Python Simple ***Forever*** *Server and C Client.*


C:\Windows\System32\cmd.exe - simple_server_pb.exe 54321
Simpl...
Version...
Current File
simple_server_pb.py
simple_forever_server_pb.py
simple_client_pb.py
Run
simple_forever_server_pb
simple_client_pb
_PB > client > simple_client_pb.py 12:1 CRLF UTF-8 4 spaces Python 3.12 (SimpleTCPIPServer_and_Client_PB)

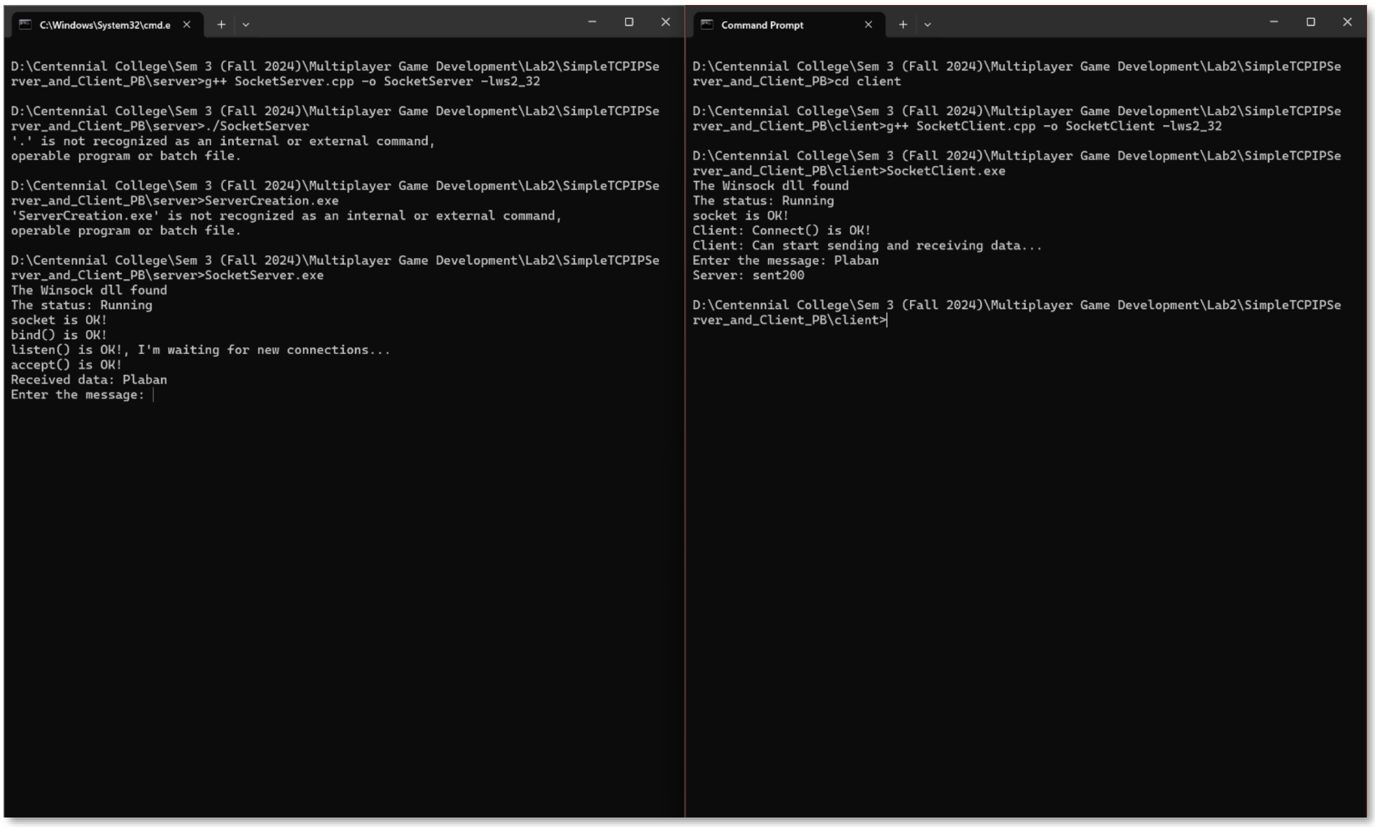
```
E:\College\Semester_5\COMP280_001_Multiplayer_Game_Development\Lab\Lab_2\Projects\C>simple_server_pb.exe 54321
server established connection with Plaban-PC (127.0.0.1)
server received 19 bytes: Hi friends !<|EOM|>server established connection with Plaban-PC (127.0.0.1)
```

```python
# simple_client_pb.py
import socket

HOST = "127.0.0.1"  # The server's hostname or IP address
PORT = 54321  # The port used by the server
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((HOST, PORT))
    # s.sendall(b"Hello, world - from client")
    s.sendall(b"Hi friends !<|EOM|>")
    data = s.recv(1024)
    print(f"Received {data}!")
```

```
E:\College\Semester_5\COMP280_001_Multiplayer_Game_Development\Lab\Lab_2\Projects\Python\S
Received b'Hi friends !<|EOM|>'!

Process finished with exit code 0
```

Fig 3.4: C Simple Server and Python Client.

4. C++

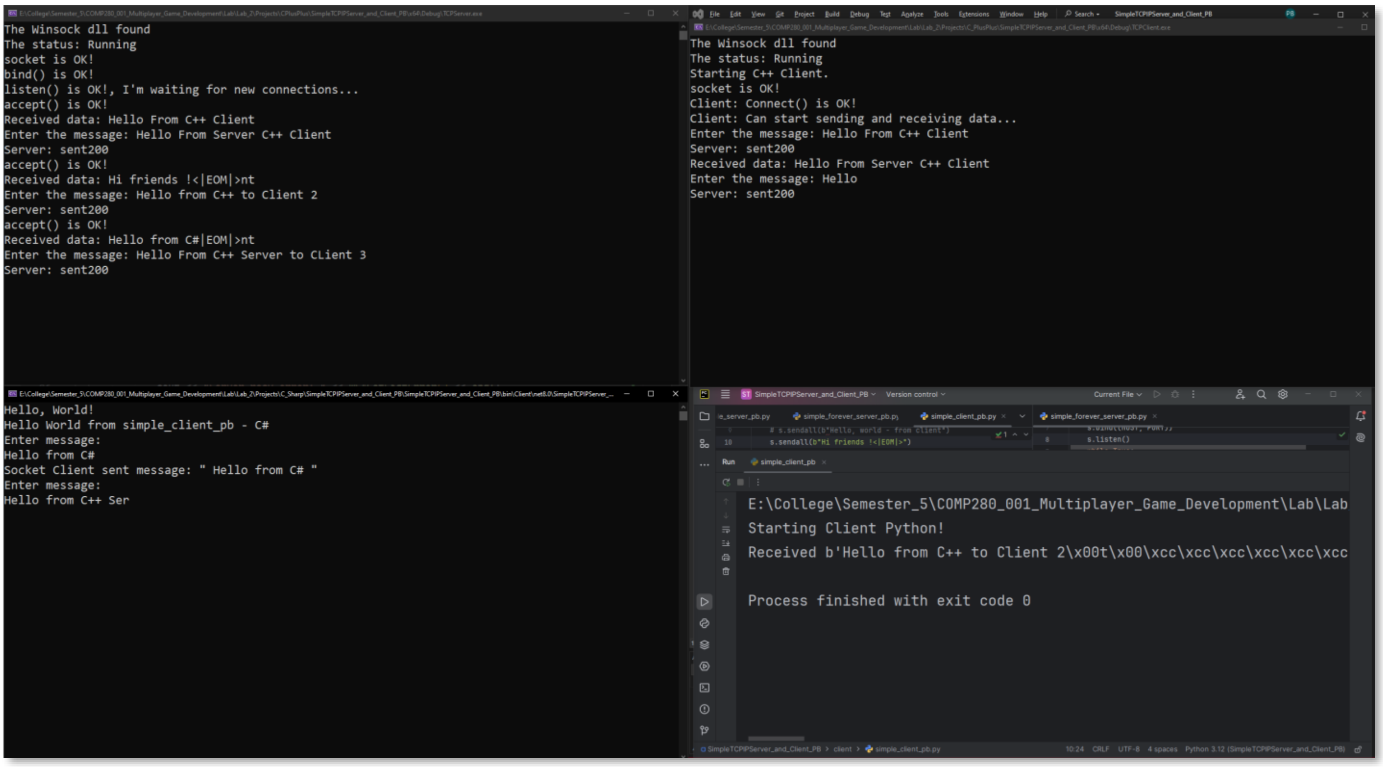


```
D:\Centennial College\Sem 3 (Fall 2024)\Multiplayer Game Development\Lab2\SimpleTCPIPSe
rver_and_Client_PB\server>g++ SocketServer.cpp -o SocketServer -lws2_32

D:\Centennial College\Sem 3 (Fall 2024)\Multiplayer Game Development\Lab2\SimpleTCPIPSe
rver_and_Client_PB\server>./SocketServer
'.' is not recognized as an internal or external command,
operable program or batch file.

D:\Centennial College\Sem 3 (Fall 2024)\Multiplayer Game Development\Lab2\SimpleTCPIPSe
rver_and_Client_PB\server>ServerCreation.exe
'ServerCreation.exe' is not recognized as an internal or external command,
operable program or batch file.

D:\Centennial College\Sem 3 (Fall 2024)\Multiplayer Game Development\Lab2\SimpleTCPIPSe
rver_and_Client_PB\server>SocketServer.exe
The Winsock dll found
The status: Running
socket is OK!
bind() is OK!
listen() is OK!, I'm waiting for new connections...
accept() is OK!
Received data: Plaban
Enter the message:
```

```
D:\Centennial College\Sem 3 (Fall 2024)\Multiplayer Game Development\Lab2\SimpleTCPIPSe
rver_and_Client_PB>cd client

D:\Centennial College\Sem 3 (Fall 2024)\Multiplayer Game Development\Lab2\SimpleTCPIPSe
rver_and_Client_PB\client>g++ SocketClient.cpp -o SocketClient -lws2_32

D:\Centennial College\Sem 3 (Fall 2024)\Multiplayer Game Development\Lab2\SimpleTCPIPSe
rver_and_Client_PB\client>SocketClient.exe
The Winsock dll found
The status: Running
socket is OK!
Client: Connect() is OK!
Client: Can start sending and receiving data...
Enter the message: Plaban
Server: sent200

D:\Centennial College\Sem 3 (Fall 2024)\Multiplayer Game Development\Lab2\SimpleTCPIPSe
rver_and_Client_PB\client>
```

Fig 4.1: C++ Server and Client.

5.Challenge



```
The Winsock dll found
The status: Running
socket is OK!
bind() is OK!
listen() is OK!, I'm waiting for new connections...
accept() is OK!
Received data: Hello From C++ Client
Enter the message: Hello From Server C++ Client
Server: sent200
accept() is OK!
Received data: Hi friends !<|EOM|>nt
Enter the message: Hello from C++ to Client 2
Server: sent200
accept() is OK!
Received data: Hello from C#|EOM|>nt
Enter the message: Hello From C++ Server to Client 3
Server: sent200
```

```
The Winsock dll found
The status: Running
Starting C++ Client.
socket is OK!
Client: Connect() is OK!
Client: Can start sending and receiving data...
Enter the message: Hello From C++ Client
Server: sent200
Received data: Hello From Server C++ Client
Enter the message: Hello
Server: sent200
```

```
Hello, World!
Hello World from simple_client_pb - C#
Enter message:
Hello from C#
Socket Client sent message: " Hello from C# "
Enter message:
Hello from C++ Ser
```

```
s.sendall(b"Hi friends !<|EOM|>")
s.listen()
```

```
E:\College\Semester_5\COMP280_001_Multiplayer_Game_Development\Lab\Lab
Starting Client Python!
Received b'Hello from C++ to Client 2\x00t\x00\xcc\xcc\xcc\xcc\xcc\xcc

Process finished with exit code 0
```

Fig 5.1: Forever C++ Server accepting messages from C++ client, C# client and Python client.