v1.1

echo_server.py:

```
class EchoServer:

    def __init__(self):
        self.addr = None
        self.port = None
        self.socket = None
        self.host = "127.0.0.1"
        self.msg = None
```

```
def receive_message(self):
    """
    Receive a TCP packet from the client.
    :return: None
    :return: the received message
    """
    pass
```

```
def send_message(self):
def send_message(self, msg):
    """
    Send the message stored in self.msg back to the client
    :param conn: a socket object
    :param msg: the message to send to the client
    :return: None
    """
    pass
```

Number_client.py

Change to inherit from echo_client.py.

```
from echo_client import EchoClient
import socket
import sys


class NumberClient():

    def __init__(self):
        self.host = "127.0.0.1"
        self.port = None
        self.socket = None

    def read_port_number(self):
        """
        Read the port number from argument, store it to self.port.
```

```
        Exit if invalid argument is provided.
        :return: None
        """
        pass

    def connect_to_port(self):
        """
        Create a socket to try to connect to a specified port,
        store the new socket object to self.socket.
        :return: None
        """
        pass

    def receive_and_print_message(self):
        """
        Receive a TCP packet from the server and print
        it out to stdout
        :return: None
        """
        pass

    def send_message(self):
        """
        Read messages from stdin, then convert to bytes and
        send out as a TCP packets
        :return: None
        """
        pass

    def run_client(self):
        """
        Run the client to send and receive messages with the TCP
server
        :return: None
        """
        self.read_port_number()
        self.connect_to_port()
        while True:
            self.send_message()
            self.receive_and_print_message()


if __name__ == "__main__":
    number_client = NumberClient()
    number_client.run_client()
```

```
from echo_client import EchoClient
import socket
import sys


class NumberClient(EchoClient):
    pass


if __name__ == "__main__":
    number_client = NumberClient()
    number_client.run_client()
```

v1.2

Add self.conn to echo_server.py and chat_server.py to accept new connection from the client.

echo_server.py

```
def __init__(self):
    self.addr = None
    self.port = None
    self.socket = None
    self.host = "127.0.0.1"
+++ self.conn = None
```

```
def listen_on_port(self):
    """
    Create a socket listens on the specified port.
    Store the new socket object to self.socket.
+++ Store the new accepted connection to self.conn.
    :return: None
    """
    pass
```

chat_server.py

```
def __init__(self):
    self.host = "127.0.0.1"
    self.client_name = None
    self.port = None
    self.socket = None
+++ self.conn = None
```

```
def listen_on_port(self):
    """
    Create a socket listens on the specified port.
    Store the new socket object to self.socket.
    :return: None
    """
    pass
```

```
def recv_client_connection(self):
    """
+++ Accept a client connection and store the new
+++ accepted connection to self.conn.
    Get and store the client name in self.client_name.
    Print the get connection message to the stdout.
    Send the welcome message to the connected client.
    :return: None
    """
    pass
```