



CMPE 472 – Computer Networks

Programming Assignment 1

Topic: Mini Weather Prediction

In this assignment, you will create a mini application that acts as a "Weather Prediction" system using socket programming. The server will randomly pick a city's weather temperature from the given Excel file, and the client will try to guess the temperature within a certain tolerance range. You can choose any development environment (e.g., PyCharm, Visual Studio Code, etc.).

Our goal is to create a connection between a server and a client on the same machine (localhost) and enable communication through sockets.

Requirements:

- Specify which port the server and client communicate on and provide the output.
- The server must listen to the client on port 8888.
- Below, you can see the general code blocks to use. Code the methods as needed.

server.py

```
def handle_request(client_connection):  
    # This method should process the client's guess  
    # and respond with the appropriate message based on the guess accuracy  
  
def serve_forever():  
    # In this method, load the weather data, randomly select a city,  
    # and wait for the client to guess the temperature of the chosen city.  
  
if __name__ == '__main__':  
    serve_forever()
```

1. Load the weather data from the attached weathers.xlsx file on the server side. This file contains city names and their recorded temperatures.
2. The server should randomly select a city from the Excel file and send a message to the client, asking for a temperature prediction.
3. Validate the client's prediction:

- I. If the prediction falls within a 10% tolerance range of the actual temperature, the server should respond with a success message.
- II. If the guess is too high or too low, provide a hint ("Higher" or "Lower").
- III. If the client makes three incorrect predictions, the server should send the correct temperature and wait for a new client connection.
- IV. If the client sends the command "END", both the client and server should terminate the connection.

client.py

```
def main():  
    # Connect to the server, receive the city information,  
    # prompt the user for a temperature prediction, and send it to the server.  
  
if __name__ == '__main__':  
    main()
```

1. The client should connect to the server and receive the weather information (e.g., "Predict the temperature for New York").
2. The client will then enter a predicted temperature and send it to the server.
3. After each guess, the client receives feedback from the server based on the accuracy of the prediction:

- I. If the prediction is correct within the 10% tolerance, the client will receive a success message.
- II. If the guess is incorrect three times, the game will end, and the correct temperature will be shown.
- III. If the client enters "END", the connection between the client and server should close, and both programs should terminate.

What to Hand In

You will submit a ZIP (or RAR) file containing:

- server.py (server-side code)
- client.py (client-side code)
- A PDF report explaining your code with screenshots.

The report should include:

- A cover page
- A table of contents
- Code explanations with relevant screenshots

Points will be deducted for poorly organized reports. Ensure the report is clearly prepared and well-structured.