COSC 3360/6310—OPERATING SYSTEM FUNDAMENTALS
ASSIGNMENT #2 FOR SUMMER 2017: PRICE A NEIGHBORHOOD

Due on Friday, July 21 at 11:59:59 PM

OBJECTIVE

You will learn to use stream sockets.

YOUR PROGRAMS

You are to write ***two*** programs:

1. A client program that will connect with your server and send it requests for the average price of homes in a given neighborhood.
2. A server program that will wait for connection requests from your client and return the average price of houses for the given neighborhood.

THE SERVER PROGRAM

Your server must start by prompting for the name of the file that contains the average house prices in the many Houston neighborhoods:

```
Acres Home, 123910
Addicks/ Park Ten, 210431
Afton Oaks/ River Oaks, 1645821
Airline Farms, 575000
Aldine Meadows, 199000
Alief, 135940
```

read it in and store it in a table. It should then prompt for a port to listen to as in

```
Enter server port number: 2468
```

It will then create a stream **socket**, **bind** it to the specified port number, do a **listen( )** to specify a maximum number of queued connection requests and loop through **accept( )** calls that will let it wait for connection requests.

Whenever the server accepts a connection request, it will receive a Houston neighborhood name and reply with the average price of houses in that neighborhood.

Should the neighborhood name ***not*** be in the table, it should reply with a ***negative value.***

THE CLIENT PROGRAM

Your client should start by prompting the user for a server host name and a server port number as in:

```
Enter a server host name: program.cs.uh.edu
Enter server port number: 2468
```

It should then create a stream **socket**, go through a loop prompting for a user nickname, **connect** to the server, send a request to the server by **writing** to the socket, **read** the server reply and print it out on the screen as in:

```
Enter a Houston neighborhood: Alief
The average price of houses in Alief is $135940
Enter a Houston neighborhood: Merced
That neighborhood is not in the table
…
```

Your client should end the loop when the user enters an ***empty string***.

HINTS

1. Please refer to "BSD Sockets: A Quick and Dirty Primer" at URL: http://www.cs.uh.edu/~paris/3360/Sockets.html or through the course Piazza page. It contains a general introduction to sockets. You can include any code from that document in your assignment.
2. Keep in mind that server and client processes read the messages byte by byte and have no way to know how many bytes they should read. The easiest way to do it is to put your messages into fixed size buffers. Both **sprintf( )** and **sscanf( )** could come handy.
3. There will never be more than 1,024 entries in an input file.
4. The input is supposed to be correct.
5. Use a *single-threaded server* to keep things simple. You will not have to not worry about zombies and can safely ignore the **fireman()** call in the primer.
6. Your client ***should*** request a new connection for each request as doing otherwise would either require a multithreaded server or prevent the server from handling multiple clients.
7. Yes, you will have to turn in two different programs, namely a client program and a server program.

This document was updated last on Monday, July 10, 2017.