159352 Tutorial – Week 3

In this exercise you will be implementing a basic HTTP server. You can use the server.py code below (available on Stream) as a starting point, or use any other language, which supports raw TCP/IP sockets such as Java.

Create a simple website with one index.html base page and at least 2 linked images. Place the web objects in the same directory as your web server program. This is becomes your document root. Extend the web server code such that it is able to respond to client requests, and provide the requested resources. It should be able to do basic error handling such as HTTP 404 - Not Found.

Optional exercise – Make your server multithreaded so that it can handle multiple requests.

Submit your server code on Stream.

```
#Here is a skeleton code you may use as a starting point.

#This is a very basic HTTP server which listens on port 8080,
#and serves the same response messages regardless of the browser's request.
#It runs on python v3
#Usage: execute this program, open your browser (preferably chrome) and type http://servername:8080
#e.g. if server.py and browser are running on the same machine, then use http://localhost:8080

# Import the required libraries
from socket import *

# Listening port for the server
serverPort = 8080

# Create the server socket object
serverSocket = socket(AF_INET,SOCK_STREAM)

# Bind the server socket to the port
serverSocket.bind(('',serverPort))

# Start listening for new connections
serverSocket.listen(1)

print('The server is ready to receive messages')


while 1:
        # Accept a connection from a client
        connectionSocket, addr = serverSocket.accept()

        ## Retrieve the message sent by the client
        request = connectionSocket.recv(1024)

        #create HTTP response
        response = "HTTP /1.1 200 OK\n\nWelcome to my home page"

        #send HTTP response back to the client
        connectionSocket.send(response.encode())

        # Close the connection
        connectionSocket.close()
```