**CPSC 441 – Assignment 3, Name: Farica Mago, UCID: 30111924**

**Prepare and submit a design document to explain what happens in your program if the method shutdown() is called while some worker threads are still executing. Describe the sequence of events that happens in your implementation once method shutdown is called. Follow the design document formatting requirements on D2L.**

If some of the worker thread are still executing, and the **shutdown()** is called, the variable **shutdown** becomes **true.**

```
    public void shutdown()
    {
        this.shutdown = true;
    }
```

When we check the condition for the while loop:

```
while(!shutdown){
                try{
                    // accepting the connection due to the request from the
client          Socket socket = serverSocket.accept();
                //socket.setSoTimeout(this.timeout);
                InetAddress inet = socket.getInetAddress();
                String add = inet.toString();
                String final_inet = add.substring(1);
                System.out.println("A new client has connected on the port
number: "+socket.getPort());
                System.out.println("The address of the client is:
"+final_inet);
                //Creating thread
                executorPool.execute(new WorkerThread(this.timeout,socket));
                }
                catch(SocketTimeoutException e)
                {
                }}
```

in the **WebServer.java**, the program will not go into the while loop and will stop accepting connections from other clients. After this the **executorPool.shutdown()** is called which means the server will complete previous requests of the client but will not accept any new Clients. Then we check if this termination process is taking longer than 5 seconds, if this condition becomes true then we command the server to terminate the **executorPool** at the same time.

```
if(!executorPool.awaitTermination(5,TimeUnit.SECONDS))
                {
                    executorPool.shutdownNow();
                }
```

After this, we close the server socket.