

## Socket Programming Project – Python HTTP Server, *Parallel Edition*

Name	Major	Primary Objective (60 pts)	Secondary Objectives (40 pts)	Total Score (100 pts)

### Evaluation Rubric – Primary Objective (60 pts possible):

*Web server can support concurrent requests from multiple clients simultaneously*

Description	Points	Score
Server successfully uses threads or parallel processes to service multiple client sockets concurrently. Readme.txt file briefly describes parallelism choice.	<b>45</b>	
FunkLoad benchmark configuration and results submitted for <i>original</i> web server and <i>new</i> parallel web server. Benchmark was <i>not</i> done to localhost.	<b>15</b>	
<b>Total (60 pts possible)</b>		

### Evaluation Rubric – Secondary Objectives (40 pts possible):

Description	Points	Score
Server supports HTTP persistent connections (socket stays open 30 seconds after a request, waiting for more requests)	<b>10</b>	
Server supports HTTP pipelined connections (multiple requests can be sent back-to-back without waiting for server first)	<b>10</b>	
Server gracefully shuts down upon receiving user CTRL-C	<b>5</b>	
Server supports HTTP HEAD requests	<b>5</b>	
Server provides the following HTTP response headers: Date, Server, Content-Length, Content-Type, Last-Modified, Expires	<b>5</b>	
Server provides verbose and silent modes of operation	<b>5</b>	
<b>Total (40 pts possible)</b>		

**Comments:**