CS100M Grading Guide: Project 1

The coded items below (e.g., c1e, s2a) indicate what a student’s solution should accomplish. Codes that begin with the letter ‘c’ deals with correctness; codes that begin with ‘s’ deals with style.

Grader: If a student’s solution does not accomplish task c1a, for example, then write the task code ‘c1a’ along with any diagnostic remarks you can give. Count the number of correctness and style errors separately. In the table below, the top row lists the possible scores (1 to 5). The next row lists the number of correctness errors corresponding to every score category. The style score is determined similarly. Enter the total score (maximum of 10) in CMS as the project score. If there are bonus questions, enter any bonus points separately in the "Bonus Bucket," separate from the project score.

Student: Read the grading guide for every project, even if you get a perfect score! Notice from the table below that we often give one or two "freebies," i.e., mistakes that don’t cost you any points. Learn from working on the project, and learn from any mistakes.

Scores

<table>
<thead>
<tr>
<th>Score</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of correctness errors</td>
<td>Not handed in</td>
<td>&gt;5</td>
<td>4-5</td>
<td>3</td>
<td>2</td>
<td>0-1</td>
</tr>
<tr>
<td>No. of style errors</td>
<td>Not handed in</td>
<td>&gt;6</td>
<td>5-6</td>
<td>3-4</td>
<td>2</td>
<td>0-1</td>
</tr>
</tbody>
</table>

0. General

(s0a) Variable names are appropriately chosen
(s0b) Code lines are properly indented
(s0c) All extraneous output is suppressed with semi-colons
(s0d) Reasonable line lengths, no horizontal scrolling
(s0e) Appropriate header comment in each script file
(s0f) No superfluous code

1. Medians of a Triangle

(c1a) Retrieves mouse input for triangle vertices’s
(c1b) Plots the triangle
(c1c) Computes the medians
(c1d) Plots the medians
(c1e) Computes the centroid
(c1f) Plots the centroid
(s1a) Code broken into appropriate commented sections

2. Newton’s Method for Square Roots

(c2a) Prompts user to input a number between 1 and 4
(c2b) The initial value of L is computed correctly
(c2c) Variables L1, L2, etc are assigned correctly
(c2d) At least four approximations are made
(c2e) All approximations and error values are output
(s2a) Header is printed, and aligns with output lines
3. Random Colors

(c3a) if statement is present for the case NumEdges > 1000

(c3b) NumEdges is reset to 1000, and warning is displayed, if needed

(c3c) The original value of NumEdges is saved

(c3d) if statement set up to handle color output

(c3e) The rand function and original N are used to compute prob.

(c3f) Red output with prob. $k/N$, black with prob. $1 - k/N$

(s3a) Uses if-else, and not two if statements, for color output