

CS100M Spring 2008 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.

Items marked with ** count as two errors. 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

- c and s stand for correctness and style; see table below.
- parts with ** next to them means that they are double the value, *** for triple, etc.

Score	0	1	2	3	4	5
#correctness errors	14+	11 – 13	7 – 10	4 – 6	2 – 3	0 – 1
#style errors	14+	11 – 13	7 – 10	4 – 6	2 – 3	0 – 1

General

- (s0a) Use meaningful variable names
- (s0b) Appropriate indentation
- (s0c) Appropriate comment header in each script/function file
- (s0d) [2* max] Write the appropriate M-file as specified
- (s0e) Use semi-colons as needed; not for use after if condition or else keyword (these are not statements)
- (s0f) Appropriate and concise comments throughout
- (s0g) Reasonable line lengths; no horizontal scrolling
- (s0h) [2* max] No superfluous code
- (s0i) No debugging output.
- (c0a) [2* max] Program successfully executes without crashing. (* for occasional, ** for persistent)

1. Platonic Solid

NOTE: There was an alternate version of this problem posted on the web for some period of time asking the students to find the area and volume of the nested solids. We will accept answers from either version of the question. The solution reflects the most recent version of the question.

- (c1a) Correctly uses the `disp` and `fprintf` commands.
- (c1b) [3* max] Correctly uses the constants P_r and P_R for a Platonic solid P (.5 * for each incorrect expression, round up to the nearest integer, at most 3)
- (c1c) [2* max] Correctly calculates P_r , P_R and E (or area and volume)
- (s1a) Displays the output neatly
- (s1b) Code broken into appropriate sections—reader friendly

2. A Cubic Function

- (c2a) Correctly asks and retrieves user's input
- (c2b) Correctly displays the coefficients using `fprintf` or `disp`
- (c2c) Correctly writes the if statement to distinguish between the 'Monotone' case and calculating the roots
- (c2d) Correctly calculates the derivative of $q(x)$
- (c2e) [2* max] Correctly calculates the roots r_1 and r_2 , and calculates $q(r_1)$ and $q(r_2)$
- (c2f) Correctly uses `fprintf` with `%10.6f` format to display r_1 , r_2 , $q(r_1)$ and $q(r_2)$.
- (s2a) Displays output in a neat fashion

3. Where did you click?

- (c3a) Correctly draws the red and blue lines over the given ranges (red between 0 and 9, blue between 3 and 8)
- (c3b) Correctly elicits a mouse click from the user
- (c3c) Correctly plots the mouse click on the graph
- (c3d) Correctly plots the red and blue lines
- (c3e) Correctly writes the if-elseif-else
- (c3f) [2* max] Correctly saves a message and uses the message in the title of the plot
- (s3a) Code broken into appropriate sections—reader friendly