

CS100M Spring 2007: Project 1 Grading Guide

The coded items below (e.g., c1e, s2a) indicate what a student's solution should accomplish. Codes that begin with the letter 'c' deal with correctness; codes that begin with 's' deal 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.
- Apply bonus for exemplary work or doing additional tasks.

Score	0	1	2	3	4	5
# correctness errors	> 9	7-9	5,6	4	2,3	0,1
# style errors	> 10	8-10	5-7	3,4	2	0,1

General

- (s0a) Use meaningful variable names
 - (s0b) Appropriate indentation
 - (s0c) Appropriate comment header in each script/function file
 - (s0d) Appropriate and concise comments throughout
 - (s0e) Reasonable line lengths; no horizontal scrolling
 - (s0f) [up to **] No superfluous code
 - (s0h) No debugging output.
-
- (c0a) [2* max] Program compiles without error. (1 * for each compiler error message up to 2)
 - (c0b) [2* max] Program successfully executes without crashing. (* for occasional, ** for persistent)

ASCII, course information and policies

- (c1a) Questions 1-2 answered correctly
- (c1b) Questions 3-5 answered correctly

(c1c) Questions 6,7 answered correctly

(s1a) reasonable formatting

(s1b) saves as a plain text file as requested

Hunt for the Line: lineHunt.m

(c2a) correctly assigns random number for y-coordinate (multiply and shift)

(c2b) properly marks the locations of the user's mouse click

(c2c) displays and uses appropriate title for the graph

(c2d) title changes depending whether you hit or missed the hidden horizontal line

(s1a) uses an appropriate marker to indicate its location

(s2a) displays appropriate information if user's point is too high or low (not something 'generic' or vague like 'miss')

Grade Cutoffs: gradeSol.m

(c3a) correctly uses if, elseif to figure out the letter grade

(c3b) correctly asks and retrieves user's input

(c3c) correctly states the corresponding letter grade to the inputted score (including those scores that could be two possible grades)

(c3d) correctly uses the error function for scores above 100 points

(c3e) correctly uses the fprintf command

(c3f) correctly uses the error function for negative scores

(s3a) clearly states in the comments the grading scheme, if the grading scheme is slightly different from website

(s3b) printed output inserts a newline at the end