

CS100J Spring 2001: Project 2 Grading Guide

Notes

- Please carefully review all notes written on your grading form and project.
- Find the codes for these notes below.
- Try to understand why you received the note so that you may avoid it on your next project.

Scores

- Bonus may be applied for exemplary work or doing additional tasks
- Let c and s be the number of correctness and style: see table, below
- For each program not included, remove one correctness and style point

category	Points					
	0	1	2	3	4	5
correctness	nothing turned in	$c \geq 22$	$11 \leq c \leq 21$	$6 \leq c \leq 10$	$2 \leq c \leq 5$	$0 \leq c \leq 1$
style	nothing turned in	$s \geq 19$	$7 \leq s \leq 18$	$4 \leq s \leq 6$	$2 \leq s \leq 3$	$0 \leq s \leq 1$

1. General

- (s1a) grading form provided for each partner
- (s1b) grading form properly filled out and signed
- (s1c) title sheet provided
- (s1d) table of contents provided (may be combined with title sheet)
- (s1e) pages numbered
- (s1f) pages properly bound
- (s1g) all work typed, except where permitted
- (s1h) lines of text/code not chopped off or misaligned
- (s1i) title block included at top of each program
- (s1j) **break** not used
- (s1k) code properly indented under flow-control structures
- (s1l) code properly indented under method structures
- (s1m) code properly indented under class structures
- (s1n) only major variables commented
- (s1o) control structures commented
- (s1p) comments for more than one line above the lines of code
- (s1q) consistent commenting style

2. Information and Policies

- (c2a) all the questions answered correctly
- (s2a) questions and answers interleaved

3. Operations

- (c3a) problem 1 correct
- (c3b) problem 2 correct
- (c3c) problem 3 correct
- (c3d) problem 4 correct
- (c3e) problem 5 correct
- (c3f) problem 5 uses scientific notation with **e**
- (s3a) output included
- (s3b) problems numbered using comments for clarity

4. User Input/Output

- (c4a) input-reader object properly initialized (students' choice of reader)
- (c4b) program prompts user for name
- (c4c) program welcomes user
- (c4d) user prompted to enter integer
- (c4e) user prompted to enter double
- (c4f) minimum value determined with condition or **Math.min**
- (c4g) program output min value in the format it was originally entered (any approach is OK)
- (s4a) if condition used to find min, only one temporary variable is used
- (s4b) program output included for case of user input of **1** and **2.0**

5. Conditions

- (c5a) input-reader object properly initialized (students' choice of reader)
- (c5b) specified variables (**weight**, **code**, **charge**) used [NOT GRADED]
- (c5c) shipping information obtained
- (c5d) **charge** properly determined with a selection statement
- (c5e) **charge** modified based on input **code**
- (s5a) only **charge** is output
- (s5b) output includes 6 cases (two codes and three ranges of weights)

6. Loops

- (c6a) brainstorming: included
- (c6b) algorithms: each step written as an instruction
- (c6c) algorithms: steps involving flow control indented
- (c6d) algorithms: no code given, except for keywords
- (c6e) program: random number between 0 and 100, inclusive, calculated
- (c6f) program: user guesses for target number
- (c6g) program: program keeps track of guesses
- (c6h) program: loop quits for out-of-bounds guess [NOT GRADED]
- (c6i) program: loop quits for too many guesses
- (c6j) program: loop quits for correct guess
- (c6k) program: output for cases after loop matches cases from loop
- (c6l) program: for out-of-bounds input, count is adjusted to be correct
- (c6m) output: session where user guess correct number
- (c6n) output: session where user quits prematurely by entering out-of-bounds number
- (c6o) output: session where user enters an illegal input
- (c6p) analysis: answered discrete and gives clear reason
- (c6q) analysis: answered guessing the "middle" value each time (halving, bisecting)
- (c6r) analysis: answered anything that seems reasonable; rate efficiency by counting number of guesses
- (c6s) analysis: compared chosen guessing-approach with binary search
- (c6t) analysis: suggested the most efficient technique
- (s6a) program: cool-sounding name chosen
- (s6b) program: pattern of preprocessing. loop, postprocessing used
- (s6c) program: named constants used for low and high bounds (**0** and **100**)
- (s6d) program: sole **while** used with selection inside body [NOT GRADED]
- (s6e) program: no nested loops, except for checking illegal types of input (which wasn't really required)
- (s6f) program: excessive redundancy avoided (more than one while, selections used instead of loop)
- (s6g) analysis: used correct grammar and spelling
- (s6h) analysis: answered questions individually, not all in one paragraph

7. Miscellaneous

- (c7a) miscellaneous
- (c7b) miscellaneous