Project 1 Grading Guide

Item no.	Description	Correctness	Style
PROBLEM 1	platonicSol.m	Total = 10	Total = 4
1	Interprets the outradius *R* for use with the inscribed shapes, and the inradius *r* for use with the circumscribed shapes.		1
2	Calculates the edge length *E* given a radius.	1	
3	Uses correct equation to calculate the volume of the solids. Circumscribed (2c), Inscribed (2c). Allow partial award in 1 point increments for small mistakes. (NOTE: as stated in the errata for *Insight*, the divisor for calculating Vol20 should be 12, not 4, i.e., Vol20 = ((15 + 5*sqrt(5))/12)*E^3;)	4	
4	For the calculation of volume, the student does not write one long equation for *V*, substituting equations for *E* as necessary. Instead, the student separates the calculation into natural equations (1s), including an appropriate variable for each (1c). i.e. The student uses the radius to calculate edge length first, and then uses the value of edge length to find the volume.	1	1
5	Indicates that the icosahedron is the circumscribed solid closest to volume of sphere.	1	
6	Indicates that the dodecahedron is the inscribed solid closest to the volume of sphere. For items 5 and 6 do not double count the penalty if a mistake was made in calculating the volumes but the indicated solids follow from the calculation.	1	
7	For the table, use fprintf (1c), use appropriate formatting sequence for each row of their results (1c), and have columns nicely lined up (1s).	2	1
8	Has all results in one table. If inscribed and circumscribed volumes appear in one table, the point is satisfied and any extraneous tables should be ignored.		1
PROBLEM 2	parallelogram.m	Total = 6	Total = 4
9	Both variables x and y are set to a random number in the interval (1,9).	3	

	Uses built-in function rand() twice, once for each x and y. (1c) Scales the result of rand() by 8. (1c) Shifts the result of rand() by 1. (1c)		
10	Correctly calculates fourth point, partial award in 1 point increments allowed for small mistakes.	2	
11	Use an if statement (1s) to correctly check (1c) if the fourth point is within (0 to 10) for x and y. This may be inclusive or exclusive.	1	1
12	Display an appropriate message with the fourth point for parallelogram drawn (1s), and for the fourth point not within bounds. (1s)		2
13	Draw the remaining lines of the parallelogram in any color or style.		1
PROBLEM 3	**myParabola.m**	**Total = 4**	**Total = 2**
14	Prompt user for input values: a,b,c,L,R.		1
15	Prompts are clear and indicate input assumptions: a~=0, L<R. No points are attached to this item, but do make a note of it to the student.		
16	Correctly compute the critical value: xc = --b / (2*a).	1	
17	Compute the min and max values of f(x) when for each of the following four cases. (2c: 1 point for at least 2 cases correct, 2 points for all correct, award at most 1 point if only one of a < 0 or a > 0 is considered.)	2	
18	Case 1: L< xc < R and a>0 eg: a=2, b=--3, c=4, L=--2, R=6 => fmin=2.875, fmax=58		
19	Case 2: L< xc < R and a<0 eg: a=-3, b=4, c=5, L=-10, R=10 min=--335, fmax=6.333		
20	Case 3: L > xc or xc > R eg: a=--3, b=4, c=--2, L=2, R=3 => fmin=--17, fmax=--6		
21	Case 4: L = xc or xc = R eg: a=1, b=4, c=--6, L=--2, R=2 => fmin=--10, fmax=6 eg: a=--1, b=4, c=4, L=--2, R=2 => fmin=--8, fmax=8		

22	Display the minimum and maximum values of f(x) on the interval (1c). Results are labeled and formatted (1s).	1	1
GENERAL			Total=10
23	Script starts with a concise comment describing the program. Function comment follows function header.		1
24	Code is sufficiently (but not excessively) commented.		1
25	Line lengths are not excessively long (80 columns). NOTE: It's ok if a couple lines are a little too long, especially if they are print statements		1
26	No extra output (debugging output) produced		1
27	Proper indentation is always used.		1
28	Use meaningful variable names. Do not overwrite MATLAB keywords.		1
29	Name important parameters as variables (constants).		1
30	No superfluous code (e.g., an empty if or else branch or a useless loop). Of course some students will have code that is awkward or unclear or inefficient. This point is specifically for not having code that does literally nothing.		1
31	Reasonably efficient code.		1
32	Does NOT put semicolon at wrong places, e.g., at the end of these lines: "if", "elseif", "else"," for","while", "function".		1
TOTAL		20	20

Penalties

P1	Student's code does not execute (or student provides a script when a function is required and vice-versa)	-1 from final score
P2	Student's code crashes or does not terminate (infinite loop) for normal cases.	-1 from final score
P3	All function headers and file names match those specified in the project description exactly. All input and output	-1 from final score

	variables should be of the correct type.	

Grade Calculation

Total Possible Correctness Points	TC = 20
Total Possible Style Points	TS = 20
Student Correctness Points	C = min(___ + 1 freebie point, TC)
Student Style Points	S = min(___ + 1 freebie point, TS)

Exceptions: If any file is missing/unacceptable, no freebie points can be applied to that file and subtract 3 style points for each missing/unacceptable file.

Student's final score	([(C/(TC))+(S/TS)] X 5) - Penalties
	(Out of 10; 1 decimal; no negative score; round to NEAREST)