Grades

Print

Add to ePortfolio

Grade Items

Grade Item	Points	Weight Achieved	Grade	Feedback
Practice Quiz	4 / 5	0 / 0	80 %	
Quiz #1	9 / 10	1.8 / 2	90 %	
Quiz #2	10 / 10	2 / 2	100 %	
Quiz #3	8 / 10	1.6 / 2	80 %	
Quiz #4	9 / 10	1.8 / 2	90 %	
Quiz #5	6 / 10	1.2 / 2	60 %	
Midterm Exam	22 / 30	7.33 / 10	73.33 %	
Quiz #6	10 / 10	2 / 2	100 %	
Quiz #7	9 / 10	1.8 / 2	90 %	
Quiz #8	0 / 10	0 / 2	0 %	
Quiz #9 (Bonus)	0 / 10	0		
Quiz #11	0 / 10	0 / 2	0 %	
Final Exam	0 / 50	0 / 17	0 %	
Subtotal for quizzes and exams		19.53 / 45	F	
CeeBot #1	9 / 10	2.7 / 3	90 %	Individual Feedback: This simple face demonstrates use of a single color and a few right angles. You might want to improve the program to better demonstrate what we have been learning in this chapter -- incorporate more colors, angles, and line patterns. An improved program might earn another point. 2nd version incorporates 3 colors and a dashed line. Partially reduced credit for partially late solution.
CeeBot #2	9.5 / 10	2.85 / 3	95 %	Individual Feedback: Works well and demonstrates use of a repeat() loop. Remember that we should be including explanatory comments in our code. For example, why should the initial alley be treated differently from the other 3?
CeeBot #3	9 / 10	2.7 / 3	90 %	Individual Feedback: Effective implementation of nested loops. We should be including explanatory comments in our code. It would be helpful to include a

				comment for each loop, for example "// once for each side" and "// once for each alley". It would be helpful to include a comment in the program explaining why the last alley on each side is handled differently.
CeeBot #4	8 / 10	2.4 / 3	80 %	Individual Feedback: The bot runs into the first ant, then the program ends with an error in the move() instruction. Perhaps the bot needs to start firing sooner. Remember that the target is moving toward the bot as the bot moves toward the target. Also, it looks like the bot will fire directly at the target. The directions call for barrage fire, which is to fire while turning. I hope you will turn in an improved solution. 2nd version: The program effectively eliminates the ants using direct fire. For this exercise, the bot should use barrage fire (firing while turning) to eliminate the targets. Barrage fire is described in the instructions for this exercise. Also, the program needs explanatory comments in the code.
CeeBot #5	8 / 10	2.4 / 3	80 %	Individual Feedback: The bot makes it through the course, but stops unnecessarily between pairs of flags. If you figure out why, and fix the program, it could earn another point. The use of repeat(15) seems odd, since there are only 8 pairs of flags. Would be better to use while() so that the program can handle any number of flag pairs, but then we would have to figure out a way to end the loop when there are no longer red and blue flags ahead. You might want to send in an improved solution for a little higher score.
CeeBot #6	10 / 10	3 / 3	100 %	Individual Feedback: The bot flies through each of the airpoints. The program includes abundant helpful comments. Good use of abs() function. Thank you for including a comment in the program explaining the +2 in the move() instruction.
CeeBot #7	8.5 / 10	2.55 / 3	85 %	Individual Feedback: The program works well and contains numerous helpful comments. In the initial ascent code, it would be sufficient to turn on the jet once before the loop and then wait until the desired altitude is achieved, rather than calling jet() repeatedly in the loop. It seems strange that the bot approaches the target while the difference in X position is greater than 20. What about differences in the Y position? Why not simply use the distance() instruction, so that the 20 units would apply as the bot approaches from any direction? Also, why does the program wait for a tenth of a second before turning toward the target? Why not turn toward

				the target, start driving toward it, and then wait for a tenth of a second as the bot advances toward the target? Though past the due date, there is still some time to turn in an improved program.
CeeBot #8	0 / 10	0 / 3	0 %	
CeeBot #9	0 / 10	0 / 3	0 %	
CeeBot ch.14 ex.4 Wasps 2 (Bonus)	0 / 10	0		
Subtotal for CeeBot assignments Subtotal for CeeBot assignments		18.6 / 27	D	
Homework 1	0 / 10	0 / 2	0 %	
Homework 2	4.5 / 10	0.9 / 2	45 %	Individual Feedback: #2: Correct. The problem was simply to break up an instruction into its component parts. #4: Correct. #5: This problem was way quite complex. The Questions & Exercises on pages 96-98 should be helpful for getting started on this problem. However, this problem has another wrinkle in that it does not ask about the state of the machine after each instruction has been executed, but between the load phase and the execute phase of each instruction. #7, #9: good #12a: The instruction at address 0 is 2655. Each instruciton takes 2 bytes, so both the 26 and the 55 make up the insturciton. Translate this instruction in to English, the same as you did for queston 7. #12b: if we know from 12a that the value 55 is being loaded into register 6, 12b is simple. #15: here again, as in question 12, we fetch two consecutive bytes of memory for every instruction. Thus, the instructions to be performed are 1202, 3242, and C000. We can look these up in Appendix C to see what they do. #34: About half of these are correct. j, for example, doesn't have enough bits in the answer. #37: AND is not the correct operation. Counterexample: input=00000000 would also produce 0000000. Note "if and only if". #46: Each frame has 1920x1080 pixels. Each pixel requires 24 bits (8 bits each for red, green, and blue). So far we have 1920x1080x24 bits. now we want to send that much data 30 times per second, so we multiply by 30 to get the number of bits per second. The we compare that data rate to the nominal data rate of USB1.1, USB2.0 and USB3.

Homework 3	5 / 10	1 / 2	50 %	Individual Feedback: #1, #3, #6: good #25: Most multitasking systems give priority to interactive processes to improve the overall responsiveness of the system. The I/O bound task uses little or no resources while waiting for the input or output operation to complete, which leaves the computer free to work on the lower-priority compute-bound processes. When the input data arrives, the operating system will generally interrupt the compute bound process so that the data can be given to the requesting process, or new data can be queued for a process which is doing output. These interactions are typically short. If the I/O bound processes needs lots of CPU time, then it's no longer an I/O bound process, but must wait its turn with other computable processes. See also the last paragraph on page 142. #31: good Social Issues #3 answer is missing.
Homework 4	9 / 10	1.8 / 2	90 %	Individual Feedback: #1:The chapter lists a variety of protocols on pages 171-173 and 189. #11: The internet is not the world wide web, which is the point of question 25. #17: good #25: OK #34: good Social issues #1: Interesting insight Soical Issues #5: expressed well.
Homework 5	6 / 10	1.2 / 2	60 %	Individual Feedback: #4: ok #5: Not an algorithm in the strict sense, because it will not be a terminating process. #6: The steps seem unambiguous to me because precise coordinates are given. The problem is that the 3rd step is not executable. #7: The original program prints the integers 2 thru 6, your program prints only the numbers 1 thru 5. When changing a while() loop to a repeat ... until loop, we generally need to change the condition to the complement of what it was before. The complement of (Count < 7) is (Count >= 7) #13: good #22: Since Current starts at 1 and increases from there, 0 would not be one of the values printed. #25: good

				#27: the termination condition is (or are) the condition(s) which will cause the loop to NOT execute another time. In the case of while() loops, the termination is the complement of the condition that keeps the loop running. In the case of repeat ... until loops, it IS the condition in the until(). #15: ok
Homework 6	9.75 / 10	1.95 / 2	97.5 %	Individual Feedback: #1, #6: ok #7, #9: very good #15: correct #19: correct, except that Python uses indentation instead of { } #23: Oops! we lost the STOP instruction #32: very good Social Issues #4: good
Homework 7	0 / 10	0 / 2	0 %	
Homework 8	0 / 10	0 / 2	0 %	
Homework 9	0 / 10	0 / 2	0 %	
Homework Subtotal		6.85 / 18	38.06 %	
Attendance	0 / 10	0 / 10	0 %	
Classroom Participation (Bonus)	0 / 10	0		
Points earned so far		44.98 / 100	44.98 %	

Points earned from graded items, adjusted to a 100-point scale