After the exam, we will move full speed into the Java phase of the course which began today. To COMBINE beginning to learn Java with reviewing for the exam, I strongly recommend you read Chapters 1 through 6 of Downey's How to Think Like a Computer Scientist, JAVA Version. For now, skim over the details specific to Java and use the book as a tool for learning the concepts more solidly. Try to translate the Java into Python!

This second reading will be easier and more informative because you now have more experience than you had when reading in ThinkCS was first assigned. BUT: 5 pts. will be awarded for writing a correct Java "Hello World" program..you will find it the above book, so memorize or copy it!

There will be a question or two that will be graded according to how many "function points" are correctly implemented by Python code you write. See my explanation under question 2 of the first midterm answer key. Care in following instructions for what a program should do is counted heavily! including print vs return!

I will also write the exam to sample the questions and topics below. They were gleaned from Chap 1-6 of ThinkCS Java, the lab, in-class and homework projects, and Kumar's book and the lecture notes.

There will be no questions on robot motion command details or Myro drawing and sound (Chapter 8). There may be a question about (linearly) normalizing a sensor value. There will be questions about Myro image processing (Chapter 9) and the programming aspects of Chapters 6 and 7. Specifically, there will be questions about the decision making and looping used in 6 and 7. Lists will be covered in general, as in Chapter 7, in the menu item chooser and in histogram calculations.

As in the first midterm, one 8 1/2 x 11 sheet of notes, writing instruments (pencils with eraser recommended), are allowed but no other equipment use is allowed. I will expect you to be prepared to correctly use the Myro image operations from the image project and labs, so include notes about them.

1. Explain, distinguish and use in programs: Input, Function arguments, Output, a Function return value.
2. Explain, show the effect of when tracing a program, and use in programs: The return statement with a return value and without a return value. State the two different things that a return statement with a return value (like return 'A Happy Thought') causes to happen when it gets executed.
3. Explain, distinguish and use in programs: Testing (as done by if statements) and Repetition (as done by Python for and while statements.)
4. Recognize, give examples of, and use in writing programs: Breaking a large task into smaller ones (at the scale of projects assigned so far.)
5. Given a Python list that might contain lists, lists of lists, lists of lists of lists, etc., where some of the lists contain numbers or strings, write Python expressions (like L[1][0][3]) to access an indicated element; and given such an expression, indicate what item it accesses.
6. Kumar's Python code in pages 161-162 of Chapter 7 for combining robot behaviors uses lists in two different ways. Given a copy of that code, identify and explain (1) what is in the list used in each way and (2) how each list is used, in terms of the behavior combination technique.
7. Debugging: Explain and/or demonstrate what to look for in response to specific Python error messages.
8. Distinguish 3 kinds of errors: Syntax, Runtime, Semantic (i.e., Logic). Identify each kind in program examples.
9. Tracing: Play the role of the computer or debugger ACCURATELY (like a professional or ideal robot) and express the results on paper. Do this accurately even when loop statements are interspersed with non-loop statements, and nested loops are combined with non-nested loops, and some variables are re-
10. Distinguish formal and natural languages.
11. Define parsing. Define syntax. (copy or memorize definition from Think Java or Python)
12. Write the Java "Hello World" program (copy from Think Java).
13. Which language has "all programs made up of class definitions"? (Python? Java?)
14. Why did Prof. Chaiken insist that some Python projects start with a main( ) function? Hint: Java is in the syllabus.
15. Grouping of statements, as in the body of typical if or while statements, is expressed by curly braces { like=like+1; this.fun(); } in Java. What expresses grouping of Python?
16. Recite the definition of "type" given in class and give examples of that definition in Python. How do types help you use a function or operation? How do types avoid mistakes? How many int values are there?
17. Why is it important to code 1.0/3.0 or 1.0/3 but not 1/3 to compute, say, how to split a pound of coffee among 3 friends?
18. What kind of thing does a variable store? What does the assignment statement change?
19. Unlike Java, we do not and cannot "declare" variables when we use them in Python. What happens when a Python program attempts to use the value of an undefined variable? (A Java variable is never undefined.)
20. Why do you think variables cannot have names that are keywords? One reason is "it is against the rules for the language," which is true. But think of reasons for that rule and write them down.
21. Why is 1+2*3 evaluate to 7 and not 9 in Python? Express in Python, without using variables, the computation that adds 1 plus 2 first and then multiplies the result by 3, to get 9. Express THE SAME computation using a variable but no parentheses.
22. This is good Python code: fred = 3; print fred * 'sleepy ' + 'cat.' What does it print? (It is bad as Java code.)
23. In Python, * (asterisk) and + (plus) are called polymorphic operators because they each do different things depending on the type of values they operate on. Compare + on numbers with + on strings. (Review Chapter 3 of Kumar.) What if you "multiply" a string by an integer?
24. Why is minute + 1 = hour illegal in Python (and Java)? What does minute + 1 == hour mean?
25. Why does the result of computing an expression have a type but the result of the computation programmed in a statement does not have a type? (Hint: Which result is a value?)
26. Given a Python program and one selected line in it (like "the fifth line"), encircle the entire Python statement that begins at that line. Encircle the entire Python expression that is on the right hand side of an assignment statement or is the conditional expression in a while or an if statement.
27. Explain, distinguish and use in programs: One if/elif/elif/ ... /else statement and a sequence of two or more if statements. Draw flowcharts for examples of these. ACCURATELY interpret program fragments that contain examples of these.
28. Explain "incremental development" and what is "scaffolding" in that context. (Read in ThinkCS.)
29. Write a correctly expressed Python function to perform a calculation for a task described in the exam for calculating something from a Myro picture, along the lines of the image project and the lab work. It should use the relevant Myro functions properly. It might make decisions based on the R, G and B intensities of pixels. Write a function to print the histogram of positive ints. a user types in, and calculate something based on the histogram list.
30. Write code to assign a Python function to a Python variable, pass the function to another function fOfFun as an argument (i.e., a functional argument), and have the fOfFun function use the function passed to it; all to accomplish a task specified in the exam.
31. Write Python code to construct a Python dictionary, put items in it with their keys, and retrieve the items for given keys. Do this for a specific example of items and keys.