As in the other exams, one 8 1/2 x 11 sheet of notes, writing instruments (pencils with eraser recommended), are allowed but no other equipment use is allowed.

In short, the questions will be based on what I expected you to learn from the projects and exercises, and much of what I expected from labs, lectures and going a little further on the web site.

1. Given the information for each particular case, draw trees of each of the following: (A) Activations of non-recursive methods calling each other, (B) activations of a recursive method, (C) graphics drawn by fractal-like recursions as in the labs, (D) diagram of directories (i.e. folders) and files for a simple web site, and (E) all subsequent move in Tic-Tac-Toe. In a Tic-Tac-Toe game tree, figure out (a) Kumar's openWins(player) - openWins(opponent) static evaluation function values and (b) the -1.0, 0.0, +1.0 scores for X and for Y from full backtracking.

2. Write an algorithm outline, pseudo-code expression or a flowchart for the person vs. computer game loop.

3. Trace Kumar's subsumption code or modifications of it given the necessary information about the robot's sensor data. This requires understanding the list of functions, the return values which are lists, the use of a function argument, etc.

4. Code in Java the main function plus two or three static methods for a specified task, where the methods must carry out specified parts of it. Some of the methods must return a value without printing and others must print without returning a value (but still returning."

5. Code in Java a class with the main function and a different class with non-static methods, and code the main function to instantiate the other class twice or so and to use the instances.

6. Write Java code to use an array and to prevent array index values derived from user input from causing an "java.lang.ArrayIndexOutOfBoundsException". Doubly nested loops, loops within loops, loops to list things and loops to find things and loops to change things. Image pixel processing.

7. Distinguish and use in programs: Input, Function arguments, Output, a Function return value.

8. 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. Trace other Java usages and elements covered in the course and projects.

9. Recognize, give examples of, and use in writing programs: Breaking a large task into smaller ones (at the scale of projects assigned so far.)

10. 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-initialized before a loop and others are not. (Heed the double loop quiz and makeup HW.)

11. Miscellaneous questions (not all need be answered) on say Rodney Brooks, Brian Kernighan, exceptions, fractals, Tim Berners-Lee, robotics intuitions from the robot phase of course (memorization of Myro and Python details will not be required)...

12. Structure of a simple web site.