

Extra practice ideas:

Program writing questions:

Your assignments can provide you with an idea of the type of code writing questions that you could see. Because you won't have as much time on the exam as you did when you worked on the assignments will be asked to implement one or two methods rather than the complete program. That means you should *be prepared to be given a framework program that you are expected to implement a solution within.*

Idea number 1: Simplified version of checkers

There are two practice assignments posted on the course web page. The first one is closest to what you may have to implement for the midterm (the second one makes for good practice for the final exam because it focuses on inheritance):

http://pages.cpsc.ucalgary.ca/~tamj/219/assignments/practice/oo_assignment/PracticeAssignment1 Checkers.pdf

Keep in mind that the version of checkers that you would be asked to implement would only have human players. Also because you haven't learned much (if anything) about recursion you wouldn't be asked to implement a feature that will allow players to "take back" previous moves.

Idea number 2: Combat simulation game

Another practice problem could involve something similar to the Game of Life (array of references to objects). However the program could allow for combat between adjacent pieces (the damage can be some fixed amount or some random value). That also means that each occupant of an array cell must an attribute such as 'health' or "hull points" depending upon whether your simulation has a fantasy role-playing theme or the theme of space combat. Again to keep it simple you can say that there will only be human players in the game (when one array element is adjacent to another the program will ask the player which – if any – entities that he or she wishes to attack).

Here are some of the classes that would have to be implemented:

- **Driver:** starting execution point
- **UserInterface:** displays prompts to the user and gets input
- **GameWorld:** similar to the **Biosphere** class, contains an array of references to objects. Empty elements can either have the appearance attribute of the array occupant either set to a space (as was the case with the Game of Life) or empty elements could simply be set to null. (In the latter case the display method would have to be carefully written so the null pointer isn't directly accessed but instead a space is displayed instead).
- "Critter equivalent" could be a "Spaceship" or "Creature" class. It contains a health/hull attribute and the ability to generate an attack on another object. Similar to a 'Critter' it will also have an appearance attribute.

Granted that you will have to fill in the blanks with the above framework (e.g., what values to use for the health and damage but these specifics probably aren't so crucial because the idea is to get practice for the exam not to implement a playable game). Again you wouldn't be asked to implement an entire program for an exam. Depending upon the level of complexity and

difficulty of the exam question you could be asked to implement various methods such as: initialize the game world randomly or based on user input, display the game world, writing attack and/or deducting damage methods just to name a few.

Idea number 3: Plague simulation (scientific and not a zombie game)

Although it is an old CPSC 231 assignment I listed it as an extra practice idea because it requires an Object-Oriented solution but also uses some of the rules of the Game of Life as a starting point so some parts of the 'rules' will be familiar to you:

<http://pages.cpsc.ucalgary.ca/~tamj/2010/231W/assignments/assignment5/index.htm>

Ideas for practice trace questions:

Take the existing programs and modify them e.g., (assuming there exists an attribute 'a') make 'a = 2' become 'this.a = 2' or 'super.a = 2'. As long as the result is syntactically valid then you can use the modified program for extra practice. Just make sure that you try your hand at a hand trace before running the program otherwise you will get nothing out of the exercise.

Other sources of ideas

Finally you can look through the recommended paper text book as well other Java textbooks you find on the library 'Safari' website:

<http://proquest.safaribooksonline.com.ezproxy.lib.ucalgary.ca/>

(In the latter case just make sure that you select generic Java textbooks, ones that list topics which we have talked about such as "NetBeans" or "Enterprise" in the title will likely be too specific).