CSEN 275: Object Oriented Analysis, Design, and Programming

Computerized Garden (Simulation in Java)

Team: 3 people

Description:

Analyze, Design, and implement a computerized gardening system. You will need to submit your Requirements and OO Analysis documents (problem statement, feature list, and use cases in the form of user stories and scenarios), OO Designs (UML diagrams), and OO Software (Java).

Requirement Analysis:

Find out what all that is needed for an automated gardening system. Resources you should utilize:

- Gardening books (Go to a library if you have no gardening books)
- Talking to a gardener
- The internet

Look through the slides for all the documents you need. The format of the documents is not dictated so you are free to use any format you wish as long as it includes all the necessary information.

Design:

Design your system. Meaning figure out all the classes and objects you will need as well as the proper relationships between them in order to get the system working. In doing so you will need to draw all relevant UML diagrams for the system. Nothing should be left out!!

Implementation:

Implement your design via the Java programming language. We will be running your program and watch your growing plan work for a given amount of time. Your TA and I will play the role of human gardeners interacting with your system's growing plan.

Some requirements:

- You must have a Java FX user interface. Poor UI designs will lose points! Resources for GUI: Last 4 chapters of the Java textbook for the class.
- You must have at least 3 Modules beyond the plants in the garden. An implementation that does not include various standalone modules will not receive a good grade! Some module examples are: watering system, heating system, pest control, etc.
- Your garden must be a big garden with many plants of various kind, insects, sprinklers, sensors, etc.
- Your system must keep a detailed log of every occurrence, interaction, etc. We will run your program and then read through the logs. If the logs are poor in quality and hard to navigate then even though your program may be working correctly, since we won't be able to determine that, we will assume that it does not work properly and

thus you will lose significant amounts of points. You will not get a chance to remedy this by explaining your log or program afterwards.

Documentation:

Please provide ample documentation for your implementation. If the TA and I can't figure out how to use your system then you will definitely lose a lot of points! You will not get the chance to showcase or verbally explain your code. Your only chance is to make the user interface as usable and simple as possible and to provide a robust help/user manual along with your implementation. We will refer to it during grading.

Also, ensure your log is highly readable. Provide a help guide/manual for your log if need be. If your logging system fails, is incoherent, or hard to navigate you will lose a substantial amount of points. You will not get the chance to showcase or verbally explain your log file either.

Presentation:

We will have very short (<10 minutes) class presentations to showcase your garden.

Notes:

- Grading will be highly subjective! If your TA or I find it difficult to use your software / read your log file or if we find either to be of very low quality, you will most definitely not receive a passing grade on the project.
- A big part of the grade will rely on the garden's ability to survive on its own. Your TA and I will run your program for over 24 hours with a monitoring system which will record if (and when) your system crashed.
    - The grade for this part will purely reside on how long your garden lasts. For example: If your program crashes soon after starting, you get a 0. So, make sure all exceptions are handled in a manner to ensure the garden continues operating no matter what!!
    - Make sure your subsystems are working properly as if your garden is all dead when time runs out then you also won't get a good grade!! We will check to make sure you haven't hardcoded it to never die even if nothing is watered, etc. Such tricks will result in a 0 grade.
    - We will give you an API to include in your code for the monitoring system so stay tuned for that later.
- Reference any material you use to create your system. Any detected paganization will result in an instant F grade on the project and perhaps even further disciplinary actions by the department! We take this most seriously!!!