CS363 Fall 2015 -- Final review page
To prepare for this final examination you should:
  • review the lectures from class
  • reread the labs
  • answer the questions in the midterm review
  • answer the questions below
  • ask questions in class during the review week (for any of the following that are unclear)

Disclaimers: Not all exam questions will come from the following. Not all these questions will appear on the exam.

Sample Questions:
1. Describe the model-view-controller pattern. Why is it useful? What are it’s drawbacks?

2. You peek over your classmate’s shoulder (who is flopping around helplessly with hopelessly broken code) and see code like this (where they are setting the left boundary of the road to collect statistics): hateProgramming.params.theLocation.gack = 5000000; You cringe, but, wanting to help, and being a polite person, hesitate over what to say. How might you tell them why this is poor programming style? I.e. what might this have to do with why their code doesn’t work? Hint: there are at least 3 flagrant violations of good practice here.

3. Some people insist that instance variables be declared private. How would that impact the code in the previous question? This, btw, is why some instructors insist they be private.

4. Why do expert programmers test classes before incorporating them into a large project? Write your response as if you are trying to convince one of your programming partners to test their code!

5. What are four debugging techniques?

6. You find yourself, late at night, having been in front of the screen for several hours, and are frustrated and confused because your code is not doing what you had hoped it would do. You have no idea what to do next. What are three things you might do to improve the situation?

7. Your instructor has claimed that you can trade coherent class structure design for algorithmic complexity. What does that mean? Do you believe it? Assuming you do, explain how that might be possible (i.e. describe the mechanism by which it simplifies the code/coding).

8. Write a Controller class that extends Thread for use in a model-view-controller style simulation. Include a running variable, methods toggleRunning(), run(), delay(), and step() (like the generic one in examplesFromClass/MVC_example/src/MVC/GenericMVC_Controller.java).

9. Write an init() method that creates and starts a Controller.

10. Why is the mapping from a real system to a model typically many-to-one?
11. People build and use models and simulation all the time. When a person uses a model for prediction, where is the simulation happening? What is it called when your predictions fail and you adjust your model (or its transition function)?

12. People's models are always incomplete and inaccurate to some degree; but they are often still useful. What is the technical adjective for such a model (i.e. which is "good enough")?

13. Ever since Lab 0 you have been putting a JPanel into a JFrame to create a view. What method do you write in the JPanel to draw your view? What are 5 things that can go wrong (i.e. that will prevent the drawing you do in that method from appearing)?

14. Incremental implementation is a software engineering technique; why does it work? Why would anyone attempt to implement a large, complex project without using this technique? How would you convince someone to use this technique (especially if you discovered that someone floundering around trying to make "anything" work in a big, ugly piece of code)?

15. Name 5 elements of good programming style (i.e. software engineering principles).

16. Now, 5 more!

17. Why are abstract classes useful?

18. What are Java interfaces used for? Give an example (i.e. write and implement one!).

19. If you have two Threads in a program, and one is running in a loop, how does the other ever get control?

20. What causes ConcurrentModificationExceptions? How do you fix them? Give an example (i.e. write code).

21. What does it mean for a simulation to be "valid"? Give a positive and negative example.

22. What are three reasons a Monte Carlo simulation might be invalid?

23. How could you use Monte Carlo simulation to validate your traffic model?

24. Write pseudocode for boolean wantsToChangeLanes() in Driver (or one of its subclasses, your choice which one).

25. Why would it make sense to introduce state to your Driver class?

26. Imagine a prototype one lane traffic simulation in which there is a line of cars at about their preferred following distance; you notice every time step every other one slows down, and the others speed up… and on the next time step the reverse occurs. This is nothing like the way people drive. What's wrong? How might you fix it?

27. Does it make sense for cars to change lanes in a single time step? Why or why not?

28. What are the state variables in your traffic simulation? How many are there?
29. Draw a Booch diagram for your traffic simulation before Scenarios were added.

30. Draw a Booch diagram for your traffic simulation with Scenarios, but don’t expand Model (as you already did it in the previous question).

31. Perhaps glaciers (and "snowball Earth")! are caused by "snow-machine Earth". As you may recall, the jet-streams are at the margin between the three cells (see illustration, from [wikipedia:jet stream]). The hypothesis is that warming equatorial waters increase the size of the Hadley Cell (and shrink the Ferrel Cell); past some threshold, the Ferrel Cell collapses entirely, as does the Polar Cell, making one giant Hadley Cell; and voila! Use your (implicit?) model of global weather to predict how warming the equatorial waters just a little (as has happened in the last few decades) might affect the Polar Jet, and thus the weather around here.

32. Write a method which is passed a byte[8] array returns a short[4] array (the first two bytes converted to the first short, and etc).

33. Write a method which is passed and byte[], an index, i, and a short, which store the high order byte in the ith slot and the low order byte in the i+1th slot.

34. What is a buffer (in computing)?

35. Given the following InodeBuffer class, write byte[] toBytes() which will return the four instance variables embedded in a byte[8] array (in the right order!).

```java
class InodeBuffer {
    short size;
    short directIndex;
    short inDirectIndex;
    short doubleIndirectIndex;

    ...
}
```
36. Write pseudo-code for storing a String as a disk file in Unix. Draw a picture of the data structure for the String, "1234567822345678323456784234567852345678xyzzy!" (use arrows, not numbers as pointers to the various blocks in memory!).