CS 235 - Final Exam Review Suggestions

last modified: 2021-12-10

- You can receive (a maximum) ***5 POINTS BONUS*** on the Final Exam if you do the following:
  - Make a **hand-written** Final Exam study sheet.
  - Submit a photo or scan of it saved as a .pdf, .png, .jpg, or .tiff to Canvas **by 3:00 pm on Wednesday, December 15** such that I can read at least some significant CS 235 post-Exam 2 material on it
  - Please let me know if you have any questions about this, and I hope it helps you in reviewing course concepts more effectively before the Final Exam.
  - You are **encouraged** to have this available as you are taking the Final Exam.
- The Final Exam will be given in Canvas.
  - It will be **available from 3:00 - 4:50 pm on Wednesday, December 15**.
  - It will be set up such that **you may only attempt the Final Exam once**.
  - That said, remember that **I will be in BSS 317 and also available in the regular Zoom CS 235 session from 3:00 - 4:50 pm on Wednesday, December 15.**
    - If you are in BSS 317, you can ask me questions directly during the Final Exam. If you are not, you can ask me questions via Zoom during the Final Exam.
    - This Zoom session will be set up so all microphones are muted, and you will only be able to chat with me.
    - **You are encouraged to LET ME KNOW about technical difficulties, so we can make provisions as needed!**
    - **And, I will have a small displayed window in the Zoom session with any typos or announcements that have come up so far during the Final Exam.**
  - The Final Exam will be set up such that **you will be shown one question at a time**,

    BUT there will be a list of question-links on the right-hand-side of the Canvas screen, and you can go back and forth between questions during that **one** exam attempt.
  - You are expected to work **individually** on the exam -- it is not acceptable during the exam to discuss anything on the exam with anyone else.
  - You may look up information from your Final Exam study sheet, on-line, or from the course textbook during the exam, but note that if you take too long looking up material, you may have trouble completing the exam during the time period.
  - I expect there will be a few multiple-choice questions, and the rest will be short-answer questions.
    - BUT given the Canvas quiz question options, for those short-answer questions, you will be shown a larger area for your answer than you will need!

  - You will be reading and writing code, statements, and expressions in this format. There will be questions about concepts as well.
  - You could be asked to write Java expressions, statements, fragments, methods, or up to and including entire Java classes and applications; read the questions carefully, so you do not give more answer than you have to.
  - You may be asked questions **about** Java, or about various aspects of Java.
  - You could be asked what a Java fragment does or means; you could be given an expression or fragment or class, and asked its value or what it does or what it would output in a given situation.
  - You could be asked to modify an expression or fragment or class, or to correct an expression or fragment or class, as well.
  - You might be asked to complete incomplete code (you could be given partial code, and asked to complete or modify or debug it in some way).
- A link to a packet of references and additional instructions - intended for use with the Final Exam - will be posted from 3:00 to 4:50 pm on the course Canvas site, linked from the Final Exam Instructions.
  - So, you can have it open in another browser window while you are taking the Final Exam.
  - This is intended both for reference and for use directly in some exam questions.
  - For exam purposes, the usual comments will be removed -- however, unless clearly indicated to the contrary, you can safely assume that this example code is otherwise working, correct Java.
- Your studying should include careful study of posted examples and notes as well as the lab exercises and homeworks thus far.
- The Final Exam is **cumulative**.
  - If it was fair game for Exam 1 or Exam 2, it is fair game for the Final Exam.
  - So, studying the review suggestion handouts for Exam 1 and Exam 2 would be wise; (they're still available from the course web page, under "Homeworks and Handouts").
- You are responsible for material covered in class sessions, lab exercises, and homeworks; but, here's a quick overview of especially important material since Exam 2.
  - However, because of the importance of becoming comfortable with the rich Java API, this is still possible to show up on the Final Exam):

    I could describe a package, some classes in that package, and/or methods/constructors and:
    - ask you declare instances of those classes,
    - write classes using those classes,
    - invoke methods of those classes, etc.
- Also still fair game: at this point, you should be able to look at a Java class (or collection of classes) and determine if it is an application or not,

...as well as being able to tell:

- what it is a subclass of,
- what interfaces (if any) it is implementing,
- what types its instances would be,
- what packages it is importing, and
- its visibility

- You also are still responsible for knowing the Java conventions discussed and also the CS 235 course Java style standards. You are also expected to follow these in your answers (**including indentation**).
  - You should use the **Formatting Practice Question** linked from the course Canvas site's Home page to practice writing formatted Java statements and fragments BEFORE Exam 1!
  - If you find you are having trouble with this, make sure to come by virtual student hours or ASK ME before Exam 2, so you won't lose points for poorly-indented answers.
  - (Yes, you will be dinged for { } not placed according to the CS 235 course style standards. The required style for these has been demonstrated in all of the course examples, and mentioned in the "Important Notes" sections on Homeworks 1, 2, 3, 4, 5, and 6.)
  - Also remember that Java is **case-sensitive**; an answer may lose points if it uses the wrong case (if a class name does not start with an uppercase letter, for example, or if you start a variable name with an uppercase letter, or if you write a Java keyword using uppercase letter(s)).
  - But, an exam question *may* specify that an opening javadoc-style comment is not required for *that* particular question's answer, for exam purposes.

Intro to JDBC

- You are responsible for those JDBC features that have been discussed in lecture and in lab, as well as for those JDBC features that have been used in posted course examples and in course assignments.
- What package do you need to import within a Java class that is going to make use of JDBC?
- Need to be comfortable with the basic steps required to connect to a database and run SQL queries from Java.
  - what object(s) do you need to be sure to CLOSE when you are done?
- Need to be able to tell what JDBC code would do; may need to be able to modify it, debug it, or complete it.
- should know when and how to use the methods `executeQuery` and `executeUpdate`;
  - given a desired SQL statement, you should be able to say which of `executeQuery` or `executeUpdate` would be more appropriate for executing it, how you could execute it, and what that call would return.
- in addition to the `Statement` class, you should also be aware of `PreparedStatement`
  - what is the difference between how you set up and execute a `PreparedStatement` as compared to a `Statement`? What must be done before each call of a `PreparedStatement`?

  - should know two major advantages of using a `PreparedStatement` over a `Statement`. When would you choose to use a `PreparedStatement` instead? When would it not matter so much?

intro to packages and jars

- What is the purpose of a Java package? How do you indicate that a class is part of a package?
  - How can another Java class make use of something in a package?
  - Given a package name, where must the classes within that package reside?
- What is the `CLASSPATH`? What is its purpose? How does Java make use of it? How can you use `-classpath` with the `javac` and `java` commands?
- What is a `jar` file? Why is it useful? What can be in a `jar` file? How can you create a `jar` file? How can you use a `jar` file?

intro to JUnit

- We discussed unit testing in Java -- you should be able to write a simple unit test method for a JUnit test class; you should know what to `import` for such a JUnit test class.
- Given a method's purpose, you should be able to write a testing method that could appear in a JUnit test class would pass if that method were written correctly, such that that testing method would pass if executed.
- You should be able to understand/read/write appropriate `assertEquals` and `assertTrue` method calls.