CS 5004 : Midterm Hybrid Synthesis

Description

Instead of a timed coding test or essay/multiple choice, this is your midterm. You have access to this at the beginning of the semester and you have until midway to submit this exam (an exact due date will be provided a few weeks before). This is intended to reduce the stress of timed tests. The trade-off is, of course, the quality I expect in your submission.

The expectation is that you will work on this as we complete each section and not that you will complete this right before it is due. If you wait until right before it is due, your score will suffer and the stress of completing a large project in a short amount of time will replace the stress of having to complete a timed test.

There will be a milestone submission at some point during the semester where I'll expect you to submit certain sections with the remainder due during midterm, so make sure to work on this as you go. More on that in class.

Take your time with this and create something useful. Be creative and try to enjoy the process. Most of all, as you complete each section, ask yourself, "if someone like me read this in the past would they better understand the topic?"

Tips:

- Your final submission should be grammatically correct, neat, organized, and professional.
- Anywhere a graphic would help your explanation, use one.
- The most common deduction on this assignment is: insufficient answer. This is a graduate level course.
- I'm open to alternate suggestions as well such as a structured website and other forms of presentation.
- Each chapter should be at least a page in length.
- I have the expectation you'll need to research to answer most questions.
- Simply restating what you find in a textbook or online source will not get you credit. I expect all explanations to be original and in your own words.
- If anything is directly copied from any source, this will be considered academic dishonesty and at the least you will receive no credit for this assignment.
- You are welcome to show me a sample chapter during my office hours and I'll explain how I would grade it.
- Your practicals have to be extremely well commented and must clearly demonstrate the concept. It's extra nice if they demonstrate the concept when run.
- For the practicals make sure to include clearly labeled code files. If you make me hunt for them, I won't grade them.

- Each essay question is 1 point. Possible point values are 0, .5, or 1
- Each response must be clearly labeled and mapped back for easy grading.
- You will very likely need to do some research on your own.

Theory Portion

Chapter 1 : Getting Started with Java

Discuss “Java.” Make sure to cover the following: explain why it was the language chosen for this course, compare and contrast it with Python and C/C++, and explain the compilation process in detail. Make sure to include a discussion of data type handling in Java.

Chapter 2: Object Oriented Design with Java

Explain classes and class creation in Java. Make sure to include a discussion of constructors, getters/setters, basic class design, the this pointer, and anything else someone would need to know to understand classes. Example code would be useful.

Chapter 3: Getting Deeper with Class Objects

Chapter 2 was about basic class setup. Now discuss classes in more detail. At least answer the following questions: how do you make class objects “do” things? What are dynamic and static variables/functions? How do you approach class design? Example code is useful here and you might want to consider doing some research and reading (in a text book not a wiki page) on your own.

Chapter 4: Unit Testing

It’s not enough to make a class, it needs to be tested as well. Explain test driven development, JUnit testing in Java, and compare driver testing to unit testing. Let me know anything else I need to make sure you understand unit testing.

~~Chapter 5: Exception Handling~~

~~Explain the different types of errors one can encounter while coding, what type of errors exception handles and the syntax involved. Make sure to discuss all possible syntax and how flow control is handled during exception handling. Make sure to include the hows, whys, and syntax of it all.~~

Chapter 6: Inheritance

Tell me enough to know you understand how inheritance works. I’d like to see at least the following: how do you build more complex classes with inheritance, how does this chaining help with building a class, what is dynamic dispatching and what does it have to do with inheritance, what are abstract and concrete classes? Finally, make sure to touch on the difference between composition and inheritance and the dangers of inheritance.

Chapter 7: Interfaces

Tell me what you know about interfaces. Make sure to thoroughly explain the difference between an abstract class and an interface. Clearly explain the purpose of an interface. When would you use an interface compared to an abstract class? Dig deep and explain to me why the restrictions were placed on an interface.

Chapter 8 : The Dangers of Inheritance

In module 11 you were introduced to your first real design chapter. Explain to me the dangers of inheritance when it gets out of control. Talk about how we can avoid those dangers. I suggest that the best way to explain this is through example.

Practical:

Create an example application(s) that uses the following concepts. You may use code snippets in your theory sections above, you may use one or multiple applications. Your code needs to be 100% original.

1. Basic Java comments and basic Java syntax
2. Data types in Java
3. Type casting in Java
4. Basic Class Design and usage
5. Java methods
6. Java static methods/variables
7. Unit Testing
8. ~~Exception Handling~~
9. Inheritance
10. Interfaces

You may create any example you like, but make sure your code is 100% original.

Required Submissables:

1. The code matching all code quality aspects
2. A concept map showing where and how you demonstrated all of the concepts above in a neat organized table.
3. Create a grading statement that uses the rubric below to show me the grade you believe you deserve. Be specific.
4. Create a code walkthrough video going through your applications explaining each in enough detail to let me know you wrote this code. If you don’t convince me, I’ll be asking you in for a one-on-one walkthrough. Video should be 5 - 10 minutes in length.

	Possible	Given		
Theory			**Theory**	
Chapter 1	5	0	Covers all topics, excellent quality, full confidence student understands concept	5 - 6
Chapter 2	6	0	Missing a topic, good - excellent quality, clear and understandable	3 - 4
Chapter 3	6	0	Missing topics, moderate - good quality, clarity questionable	1 - 2
Chapter 4	6	0	Part are all of the answer appears to be a direct quote or copy from an existing text	0
~~Chapter 5~~ Chapter 8	6	0		
Chapter 6	6	0		
Chapter 7	6	0		
Practical			**Practical**	
Concept 1	4	0	Concept clearly mapped, understanding clearly demonstrated, high complexity	4 - 5
Concept 2	5	0	Concept clearly mapped, understanding somewhat demonstrated, moderate to high complexity	2 - 3
Concept 3	5	0	Concept clearly mapped, understanding unclear, low complexity or quality	1
Concept 4	5	0	Code appears to be from on online example	0
Concept 5	5	0		
Concept 6	5	0		
Concept 7	5	0		
Concept 8	5	0		
Concept 9	5	0		
Concept 10	5	0		

Not included in total possible:		
Claimed extensions	20	0
Creative or went above and beyond	20	0
Code does not compile	-100	0
Code quality	-50	0
Incorrect submission or late	-20	0
Concept map not provided	-100	0
Walkthrough Video not provided	-100	0
Other	-100	0
TOTAL POINTS POSSIBLE out of 100	90	0