Midterm Review

CSCI 4448/5448: Object-Oriented Analysis & Design

Lecture 22

The Midterm

- Posted by **Saturday 10/15 at noon**
- 100 Points
- Available until **Thursday 10/20 at 8 PM**
- On Canvas, mechanically similar to a Quiz but…
  - Available for 3 hours after opening
  - Only one attempt
  - Individuals with special accommodations will be connected to an alternate test on Canvas with a 6 hour window
- Different types of typical on-line questions
  - Matching, T/F, multiple choice, multiple guess, **essay questions**
- **YOUR WORK AND ONLY YOUR WORK**
  - **Honor Code violations will be investigated and punishment assessed**

Midterm and Quizzes

- I am not directly using quiz questions in the exam
- Having said that, I may duplicate in whole or in part questions from the quizzes as I create the exam, simply because I have a topic I want to cover

- I would look at the quizzes only to identify any areas where you may have less clarity on a concept to focus your study

Midterm and Class Exercises

- Much like the quizzes, I don’t plan to take anything specific directly from class exercises
- However, there are common concepts there that may come up

- I would review the class exercises with a view to identifying anything you’re not comfortable with or need notes on

Midterm and Readings

- At this point in the lecture cycle we have been following the topics in the Head First Design Patterns book
- We will have covered material from Chapters 1 to 9 in that book
- I have supplemented and summarized the book materials in lectures
- I will only use the book to pull exam questions if I also covered it in a lecture

- For exam study, I suggest you refer to the book if you have questions regarding topics we reviewed in class that you'd like to get some clarity or examples from from the book discussions

Midterm and Lectures

- The exam questions will come primarily from the lectures, class exercises, and things we have discussed in class
- This is where I would focus studies and review

- The following slides will look at the lectures and topics covered in the exam

Lectures Covered on Exam

- L4 OO Paradigm
- L5 OO Fundamentals
- L8 UML
- L10 TDD
- L11 Design Patterns/Strategy
- L12 Observer
- L13 Decorator
- L14 Factory
- L15 Conceptual Modeling
  - Just CRC cards if anything
- L16 Singleton, Object Pool
- L17 Command
- L18 Façade/Adapter
- L19 Expanding Horizons
- L21 Template
- L23 Iterator/Composite

- Not Intros (L1-L3)
- Not Java, Python, Git (L6, L7, L9)
  - Doesn’t mean I won’t make you read code, but no writing
- Not Problem-Solution (L13b)

UML for Class Diagrams

- Reading and interpreting diagrams, no diagram creation on exam
- Key elements to know
  - Inheritance
  - Multiplicity
  - Association/Reference
    - Single arrows show = 1-way reference
    - No arrows or both arrows = 2 way reference
  - Self-association
  - Aggregation, Composition, Existence Dependency
  - Qualification
  - Interface (labeled class or pin/socket)
  - Abstract Class (labeled)

- Other UML: Use Cases/WAVE, Sequence, State, Activity

Patterns

- Strategy
- Observer
- Decorator
- Simple Factory
- Factory
- Abstract Factory
- Singleton
- Object Pool
- Command
- Null Object
- Adapter
- Façade
- Template
- Iterator
- Composite

- Know the patterns, understand their UML representations
- Know how they're used, how to differentiate them, principles involved, etc.

Principles (from Patterns/Lectures)

1. Encapsulate what varies
2. Favor composition (or delegation) over inheritance
3. Program to interfaces, not implementations
4. Strive for loosely coupled designs between objects that interact
5. Classes should be open for extension, but closed for modification (Open/Closed Principle)
6. Depend on abstractions, do not depend on concrete types (Dependency Inversion Principle)
7. Only talk to your friends (Principle of Least Knowledge)
8. Don’t call us, we’ll call you (The Hollywood Principle)
9. A class should only have one reason to change

PLUS OO Basics: Abstraction, Encapsulation, Polymorphism, Inheritance

Know the principles (& basics), what they mean, how they impact designs

Exam Notes/Strategy

- Some questions are based on knowing definitions, some are programming related, some are on how terms relate to each other
- You will not be asked to write code, but you will have to read code and recognize what's happening
- You will not be asked to draw UML diagrams, but you will need to be able to interpret them

- Find a quiet spot where you won't be interrupted
- It is open notes - have the lecture slides handy (all in Canvas Files/Class Files)
- The exam is targeted to be about an hour to 90 minutes long, you'll have three hours to take it, but…
- Manage your time
- Skip over anything you may struggle with and knock out the easy ones

- You'll do fine!