Exam Prep

Table of contents

1 Important Topics (Relevant for Exam)

1.1 Week 1

1.1.1 Lecture 1 Introduction

This is the least like a traditional lecture so I will highlight some important concepts you need to know from this lecture (unlike the Intro to CS, I will not highlight specific slides as everything we have covered is fair game for the exam).

- Software Complexity (discussion within the heading The Paradox of Simple Rules and Complex Outcomes)

- Software Development Life cycle
  - Waterfall model
- Understand the overview of the topics
  - OOA/OOD Definatation
  - UML
  - Design Pattern
  - Code Smells and Refactoring

Note: You do not need to remember anything from the Course Overview section :)

1.1.2 Lecture 2 OOP Recap (Spriha's Lecture)

- Objects and Classes (everything that Spriha covered)
- Abstraction
- Inheritance
- Polymorphism
- Encapsulation

1.2 Week 2

1.2.1 Lecture 1 & 2 OO Design Principles

- Object-Oriented Analysis and Design
- Quality Attributes
  - Functionality
  - Reliability
  - Usability
  - Efficiency
  - Maintainability
- Gathering Requirements
  - Functional and Non-Functional Requirements
- Use cases (the textual format)
  - MoSCoW Prioritization
- Analysis

  - Identifying Objects and Classes
  - Identify Relationships and Interactions
  - Different types of relationships
  - Object interaction analysis
    * CRC Card

- Design
  - DRY (Don't Repeat Yourself)
  - KISS (Keep It Simple, Stupid)
  - YAGNI (You Aren't Gonna Need It)
  - Separation of Concerns
  - Principle of Least Astonishment
  - Law of Demeter
  - GRASP Principles
  - SOLID Principles

1.3 Week 3

1.3.1 Lecture 1 Refactoring and Code Smell

- Refactoring
  - When to refactor
  - Principles of Refactoring

- Code Smells
  - Types of Code Smells (you need to know what the code smell is and the associated refactoring technique)
  - Bloaters
    * Long Method
    * Large Class
    * Long Parameter List
    * Primitive Obsession
    * Data Clumps
  - OO Absuers
    * Switch Statements
    * Alternative Classes with Different Interfaces
  - Change Preventers
    * Divergent Change
  - Dispensables
    * Duplicated Code

        * Comments
    - Couplers
        * Feature Envy
        * Message Chain

1.3.2 Lecture 2 UML

The most important part of UML chapter is being able to convert a piece of code or problem statement into appropirate UML diagrams.

- Basics of UML (background and use in OOD/M)
- Different Types of UML Diagrams
- Use Case Diagram (this is different from the use cases you looked at during W2L1)
- Class Diagrams (you should know everything within Class diagrams from our slides)
- Sequence Diagram
- Package Diagram
- State Diagram
- Activity Diagram
- Deployment Diagram

1.4 Week 4

1.4.1 Lecture 1 UML

See the topics in Week 3 Lecture 2 UML

1.4.2 Lecture 2 Design Pattern

Everything within Design Pattern is fair game for your final exam. The important thing is that you can understand where to apply which pattern. You should also be able to write small code snippets (similar to the level in lecture slides) in your answers demonstrating that you can implement the choosen design pattern. Additionally, you should also be able to draw appropriate UML diagrams to represent the design choices for each design pattern.

- Factory Method Pattern
- Abstract Factory Pattern

- Simple Factory Pattern
- Singleton Pattern

1.5 Week 5

1.5.1 Lecture 1 Design Pattern

- Decorator Pattern
- Adapter Design Pattern
- Facade Pattern
- Proxy Pattern

1.5.2 Lecture 2 Design Pattern

- Observer Pattern
- Strategy Pattern