Northeastern University
CSYE 7230 Software Engineering
Fall 2024
Midterm Exam Guide

Midterm Supporting Skills

- Applying what you have learned to solve a problem, propose a solution, and address questions
- Solidify your understanding of Software Engineering main concepts
- Solidify your technical and problem-solving skills

Preparation Checklist

- Review the material we covered in class, notes, readings, books chapters, etc.
- Midterm material is Larman's book e/3 chapters **1**, **2**, **3**, **4**, **5**, **6**, 7, **8**, **9**, **10**, **11**, **12**, **13**, **14**, **15**, and **16**, slides of weeks number **1**, **2**, **3**, **4**, **5**, **6**, and **7,** GRASP is not included.

Format of Exam

- You will have short answers, long answers, and challenging type of questions
- You may get asked to:
    - Explain concepts and support with examples
    - List the benefits or types of a concept
    - Interpret a UML diagram and answer related questions
    - Reverse engineer Java code to UML diagrams
    - For a given set of features, analyze and elicit system requirements using Agile UP process
    - Model a system from a given vision statement by writing a happy scenario, sketching a domain model, a class diagram, a sequence diagram, and a high level skeleton implementation using Java
    - Print your name on your exam sheets' front page
    - Hand your exam sheet at the end of the exam
- Be creative, give smart and concise answers, don't memorize text but rather answer questions based of your knowledge and understanding of the introduced software engineering concepts
- Good Luck!

Grading

Your solutions will be graded on your demonstration of how well you understand the Software Engineering topics we have covered so far in class, and how well you understand and incorporate topics from among the following list

Exam Study Guide Topics

- Agile Model
- Agile Changeability
- Agile Feedback
- Agile Iterative
- Architectural Tiers
- Associations
- Domain Model
- Evolutionary Process
- Feedback
- Iterative Process
- Logical Architecture
- Logical Layers
- Lower Representational Gap (LRG)
- Model-View Separation Principle
- No Silver Bullet and Software Production Problems
- Object-Oriented Programming
- OO Abstraction
- OO Encapsulation
- OO Hierarchy
- OO Modeling
- OO Analysis and Design (OOAD)
- Operation Contracts
- Requirement Analysis
- Reverse Engineering
- Software Class
- Software Complexity
- Software Conformity
- Software Invisibility
- Software Interface
- Software as a Layered Technology
- Software Object
- Software Package
- Software Process Model
- System Sequence Diagrams
- Iteration Time-Box
- UC (Use case)
- UC Black-box
- UC Boundary
- UC Essential Style
- UC Extensions (Alternate Flows)
- UC External Services/Actors
- UC Formats
- UC Main Success Scenario (Happy, basic)
- UC Actors and Types
- UC Stakeholders and Interests
- UML Diagrams
- UML Interaction (Dynamic)
- UML Static (Structural)
- Unified Process (UP)
- UP Artifacts
- UP FURPS+
- UP Phases
- UP Principles
- UP Supplementary Specification
- UP Glossary
- Waterfall Model