

Northeastern University
CSYE 7230 Software Engineering
Fall 2024
Dec 12, 2024, 4:30-7:30pm
225 Terry Ave #216
Final's Exam Guide

Final Exercises Important Skills for Software Engineers

- 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. The final material is Larman's book e/3 chapters **17, 18, 19, 20, 21, 25**, and **26**. Slides of week number **9, 10, 11, 12, 14**, and **15**.
- Study with one or more classmates, quiz each other on the list of topics

Format of Exam

- You will have short answers, long answers, and challenging type of questions
- You will be asked eleven questions; the questions will be along these points in order:
 - 1.** The UP artifacts you have built for **Swishy!** and their evolution in each phase. That would include UP phases, principles, activates, models, and model elements.
 - 2.** The artifacts you have built for **Swishy!** and their effect and relationship on/with each other.
 - 3.** Visibility
 - 4.** Refactoring
 - 5.** Pragmatic Programmer
 - 6.** Modeling and decomposing types on the Design of Objects
 - 7.** Code to refactor and write a unit test using Java `assert` keyword/function
 - 8.** The relationship between SOLID, GRASP and GoF patterns

9. Coding a singleton pattern

10. Coding an observer pattern

11. Write a sequence diagram for a given scenario, support your design choices with the right design pattern and Java Skelton-like code for a class

- 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 list on the following list

Exam Study Guide Topics

- | | | |
|---------------------------------------------|--------------------------------------|-------------------------------------------|
| • Visibility Types | • Refactoring Extract Method | • High Cohesion Pattern |
| • Use Cases & Scenarios | • Refactoring Extract Constant | • GRASP, GoF, SOLID |
| • Use Case realizations | • Refactoring Activities | • Glossary/Data Dictionary |
| • Use case Actors | • Refactoring | • Factory Pattern |
| • UP Phases | • Pure Fabrication Pattern | • Façade Pattern |
| • UP Artifacts | • Protected Variations Principle | • Domain Model |
| • Unit Testing | • Principle of Least Surprise | • Designing Objects with Responsibilities |
| • Unified Process (UP) | • Programmer Pragmatic Attitude | • Design Pattern |
| • UML Sequence and Class Diagram | • Polymorphism | • Design Model |
| • UI, Domain Layer | • Pluggable Software | • Controller Pattern |
| • Test-Driven Development | • Pass Aggregate Object | • Composite Pattern |
| • System Sequence Diagrams | • Operation Contracts | • Code Smells |
| • System Event (Operations) | • Observer/Publish-Subscribe Pattern | • Code Reuse |
| • Supplementary Specification | • Model-View Separation Principle | • Behavioral Decomposition |
| • Subsystem | • Lower Representational Gap (LRG) | • Animation Principle |
| • Strategy Pattern | • Low Coupling Principle | • Adapter Pattern |
| • Singleton Pattern | • Loose Coupling | • Single Responsibility Principle |
| • Single Access Point | • Logging | • Open Closed Principle |
| • Separation of Concerns | • Lazy & Eager Initialization | • Liskov Substitution Principle |
| • Responsibility-Driven design (RDD) | • Information Hiding | • Interface Segregation Principle |
| • Responsibility Delegation | • Information Expert Pattern | • Dependency Inversion Principle |
| • Requirement Analysis | • Indirection Pattern | |
| • Representational Decomposition | • IDs to Objects | |
| • Refactoring Introduce Explaining Variable | | |