**Mid-Term #1 - Study Guide**

**What is covered by Mid-Term Exam #1 ?**

Below is a list of the major topics we have covered so far this quarter.

All reading, lectures, homework, assignments, quizzes, and teamwork are great resources for reviewing for the mid-term. Up to and including L7, HW1, Q2 and TW3B.

**How can you prepare for Mid-Term Exam #1 ?**

Should you be prepared to summarize facts about the topics we have covered?

No, this is not what you should expect.

All the questions are essay questions, and you will be asked to analyze the topics we have covered, and compare, contrast, or discuss how they are relevant, useful or not, and how to apply them.

You won't have to draw a diagram based on a case study. You may have to interpret information from a diagram or a case study.

**Topics we have Covered**

* The role of the software architect - who, what skills, what do they do, how they do it
* What is software architecture? How do we evaluate its "goodness"?
* The design process - steps, strategies, principles, role of experience
* API models - different API models, applicability to different problem domains (added)
* Design patterns, e.g. Gang of Four - types, when they make sense, when they don't, strengths, limitations, motivation,
* SOLID design principles - when do they apply? are they all still relevant? what happened to the ones that may not be relevant any longer?
* Patterns vs Frameworks
* Software architecture and architecture - analogy to buildings, patterns
* VIOLET - case study, lessons, design patterns that apply
* Programming Languages - history, influences, applicability, suitability
* Analysis Models - nouns, verbs, rules, representations
* Analysis Patterns - conceptual, accountability, principles, key diagrams
* What you have learned getting started on the Individual Project
* What you have learned getting started working as a team on the Team Project

What won't be covered on Mid-Term Exam #1

* Anything we haven't covered yet, i.e. things after the Module for Week 4
* Non-Functional Requirements - we have covered these a little, but not in the depth that we will.
* Architectural Styles - when they make sense for a scenario, when they don't, limitations, strengths
* Architectural Patterns - styles vs patterns, idioms, tools, strategies