SE 211
Software Specification and Design II

Course Overview

Course Description

- This course focuses on software design.
- It provides an introduction to the essential concepts employed by software engineers who design large-scale, software-intensive systems in a professional environment.
- The course takes a top-down approach -- from software architecture to detailed design.
- This course introduces well-known architectural styles and detailed design patterns.

Course Outcomes

Upon successful completion of this course, a student will be able to:

- Understand software design from an engineering perspective.

Course Outcomes

Upon successful completion of this course, a student will be able to:

- Understand software design from an engineering perspective.
- Understand the role of software architecture within the software design phase.

Course Outcomes

Upon successful completion of this course, a student will be able to:

- Understand software design from an engineering perspective.
- Understand the role of software architecture within the software design phase.
- Understand the benefits associated with using different architectural patterns.

Course Outcomes

Upon successful completion of this course, a student will be able to:

- Understand software design from an engineering perspective.
- Understand the role of software architecture within the software design phase.
- Understand the benefits associated with using different architectural patterns.
- Understand the role of detailed design within the software design phase.

Course Outcomes

Upon successful completion of this course, a student will be able to:

- Understand software design from an engineering perspective.
- Understand the role of software architecture within the software design phase.
- Understand the benefits associated with using different architectural patterns.
- Understand the role of detailed design within the software design phase.
- Become familiar with detailed design tasks.

Course Outcomes (cont'd)

- Understand how to document a software design.

Course Outcomes (cont'd)

- Understand how to document a software design.
- Understand the importance and role of design patterns in detailed design.

Course Outcomes (cont'd)

- Understand how to document a software design.
- Understand the importance and role of design patterns in detailed design.
- Understand how to address user interface design within the software development life cycle.

Instructor

- Dr. Filippos Vokolos
- **E-mail:** fvokolos AT drexel.edu
- **Phone:** (215) 571-3935
- **Office:** 3675MK Room 1149
- **Office Hours:**
  - Monday 10:00AM – 11:00AM
  - Wednesday 10:00AM – 11:00AM

Teaching Assistant

- Mr. Nikhil Parakh
- **E-mail:** np657@drexel.edu
- **Office Hours:**
  - Tuesday 4:00PM – 6:00PM

Text

- Carlos E. Otero, *Software Engineering Design: Theory and Practice*, CRC Press, 2012.

Grading

- Attendance (10%)
- Assignments (40%)
- Midterm Exam (25%)
- Final Exam (25%)

Grading (cont'd)

- Each student's final grade will be calculated by adding the scores for the Assignments, the Midterm Exam, and the Final Exam.
- A corresponding letter grade will be assigned based on the students calculated score as follows:
  - [95-100] A+ [90-94] A [85-89] A-
  - [80-84] B+ [75-79] B [70-74] B-
  - [65-69] C+ [60-64] C [55-59] C-
  - [50-54] D
  - [00-49] F

Grading Rules

- All late work will receive a reduced grade, as specified in the assignment description. No extensions will be given beyond the end of the term.
- There will be no extra credit assignments.

Academic Integrity

- The university's Academic Integrity Policy is in effect for this course.
- Also, please refer to the CS Department Academic Integrity document under "Course Information" on the Bb course page.
- Violating the Academic Integrity Policy is a serious offense.
- Penalties for violating Academic Integrity can range, depending on the nature of the violation, from not receiving credit for a specific assignment to receiving an F for the course and be reported to Student Condut.