

Object-Oriented Modeling and Design

Catalog Description:

Modeling and design of complex systems using all views of the Unified Modeling Language (UML). Most effort will be in the problem domain (defining the problem). Some effort will be in the solution domain (producing hardware or software).

Prerequisite(s): ECE 175 or C SC 127A.

Instructor: Sherilyn Keaton

Email: keatons@email.arizona.edu

Office & Hours: ENGR 253 MW 1:30 – 3:00 PM, or by appointment

Website: Desire2Learn will be used for the class website (<http://d2l.arizona.edu>)

Class: MWF 12:00 - 12:50, AME 212S

Textbook: Arlow, J., and Ila Neustadt, UML 2 and the Unified Process : Practical Object-Oriented Analysis and Design, Second Edition, Addison-Wesley (Pearson Education Inc.), 2005.

Software: We will use Enterprise Architect (www.sparxsystems.com.au) this semester. All students will be required to do homework and project using Enterprise Architect. I will provide software and installation instructions when we need to start using the software.

Grading: Regular grades are awarded for this course: A B C D E.

- 40% Two (2) Exams
- 20% Final Exam
- 10% Homework
- 10 % Quizzes (bi-weekly)
- 20% Design Project (2 person team)

Reference:

- Craig C. Larman, Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd Edition, Pearson Education, Inc. 2011 (ISBN: 0-13-148906-2)
- Rumbaugh, J., Jacobson, I. and Booch, G., The Unified Modeling Language Users Guide, Addison-Wesley, Second Edition, Addison-Wesley. 2005.
- Rumbaugh, J., Jacobson, I. and Booch, G., The Unified Modeling Language Reference Manual, Second Edition, 2005.
- Fowler, M. and Scott K., UML Distilled: A Brief Guide To The Standard Object Modeling Language, Addison-Wesley, 2000.
- Jacobson, I., Booch G. and Rumbaugh, J., The Unified Software Development Process, Addison-Wesley, 1999.
- Cockburn, A., Writing Effective Use Cases, Addison-Wesley, 2001.
- <http://www.uml.org/>

Course Objectives:

Students will understand how to develop models of systems, which may contain software and non-software components, and how to represent these models using the Unified Modeling Language (UML). Students will be able to design systems (software) starting from the business and requirements model (via use cases), to the analysis model, to the design model, to the implementation model and finally to the operational model.