

## COURSE DESCRIPTION

**Department and Course Number:** CSCI 387

**Course Title:** Software Design and Development

**Current Catalog Description:** Study of techniques for the construction of large, complex software systems, including project management, requirements analysis, specification, design, development, testing, documentation, deployment, and maintenance. Students develop software systems in a group structure that simulates an industrial setting.

**Total Credits:** 3 hours

**Coordinator:** Pamela B. Lawhead, Associate Professor of Computer and Information Science. Described version was developed by Xiaojun Qi, Visiting Assistant Professor of Computer and Information Science.

**Textbook:** Stephen R. Schach, *Classical and Object-Oriented Software Engineering with UML and Java*. McGraw-Hill, 1999. ISBN: 0-07-230226-7.

Martin Fowler and Kendall Scott, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley, 2000. ISBN: 0-201-65783-X

**Other required materials:** None

**References:** <http://www.cs.olemiss.edu/~xqi/SoftwareEngineering.html>

**Course Goals:** The topics covered include review of problem-solving concepts, software development process, software requirements and specifications, verification, and validation. The emphasis is on the application of theory. This course will give students hands-on experiences to exploit seven stages of the software life cycles to develop medium-scale software as a team.

**Prerequisites by Topic:** Students should have a background in object-oriented programming, algorithms, data structures, programming languages, and systems similar in the courses CSCI 111, 112, 211, and 223. Knowledge of discrete mathematics would be helpful. Use of the Java or C++ programming language will likely be required in any programming exercise.

### Major Topics Covered in the Course:

1. Introduction to the Software Life Cycle
  - 1.1 Scope of Software Engineering
  - 1.2 The Software Process
  - 1.3 Software Life Cycle Models
  - 1.4 Teams and the Tools
  - 1.5 Testing
2. Object-Oriented Software Engineering
  - 2.1 Rational Unified Process
  - 2.2 Use Cases
  - 2.3 Introduction to Objects
  - 2.4 Introduction to Object-oriented Analysis and Design
  - 2.5 Examples of Use Case
3. The Phase of the Software Life Cycle
  - 3.1 Requirement
  - 3.2 Specification
  - 3.3 Object-Oriented Analysis
  - 3.4 Design
  - 3.5 Implementation
  - 3.6 Integration

- 3.7 Maintenance
- 3.8 Metrics and Examples

**Laboratory projects:** Two projects in teams of 3 or 4 students. The primary project was to develop an online video store rent/purchase system.

**Estimate of ABET/CAC Category Content:**

	CORE	ADVANCED		CORE	ADVANCED
Data Structures	_____	_____	Computer Organization and Architecture	_____	_____
Algorithms	_____	_____	Concepts of Programming Languages	1	_____
Software Design	2	_____		_____	_____

**Oral and Written Communications:**

Every student is required to submit at least 4 written reports (not including exams, tests, quizzes, or commented programs) of typically 10 pages and to make 2 oral presentations of typically 5 minutes duration. This includes only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

**Social and Ethical Issues:**

Not a significant focus of this course.

**Theoretical Content (Foundations):**

Not a significant focus of this course.

**Problem Analysis:**

The student analyzes a problem such as an online video rent/purchase system.

**Solution Design:**

The student develops a model, described with the Unified Modeling Language (UML), to do analysis and express designs and then uses an appropriate programming language (e.g., Java, Perl, PHP, ASP or VBscript) to implement the design.