Course Description
Methodical development of large software systems. Topics include: models, project life cycle, requirements and specification, structure charts and design criteria, incremental implementation, software metrics. Students will participate in the realization of both group and individual software systems. Prerequisite: CS 3481.

Text
Object Oriented Software Engineering: Using UML, Patterns, and Java

Course Objectives
- Modeling with UML and project organization
- Requirements Elicitation and Analysis
- System design and Object design
- Mapping models to code and implementation
- Project integration and testing
- Project management and software life cycles

Evaluation
- Home work, Class work/preparation/participation 20%
- Course Project 40%
- Midterm 15%
- Final Exam 25%

These percentages represent guidelines and may vary during the semester. Persons with special attendance-related issues (e.g., military, athletic) or disability issues must speak with instructor in advance. More than 3 absences are grounds for failing the course.

Academic Integrity
Students are expected to abide by Appalachian’s Code of Academic Integrity. Violations will be handled as specified by that code.

Teaching Philosophy
The course will model a corporate environment; goal setting, performance review, self-motivated learning, status meetings, arbitrary management decisions, work product deliverables, and training workshops.

Consider yourself a “junior engineer” software developer!
Some “lectures” will introduce material, others will be Q/A-type sessions on material introduced outside of “lecture” via textbook reading and/or podcasts. Some “lectures” will be project-oriented including status meetings, demonstrations, training sessions, and development.

Development Technology
The course project may require the Microsoft .NET development environment.