

## EE 467 Object-Oriented Software Engineering

**Credits:** 3

**Categorization of credits:** engineering topic

**Instructor(s):** Yingfei Dong. Revised Jan. 8th, 2021.

**Text Book and Other Required Materials:** “Object-Oriented Software Engineering”, 2nd, Edition, B. Bruegge and A.H. Butoit, Prentice Hall, 2004.

**Designation:** Elective

**Catalog Description:** EE 467 Object-Oriented Software Engineering(3). Introduction to the advanced techniques for designing, implementing, and testing computer software with a particular focus on using object-oriented design, analysis and programming to produce highquality computer programs that solve non-trivial problems.

**Pre-and Co-requisites:** Pre-requisites: EE367 – Computer Data Structures and Algorithms, or instructor consent.

**Class/Lab Schedule:** 3 lecture hours per week.

### Topics Covered:

- What is Software Engineering. (3 hours)
- Software Engineering - life cycle: analysis, design, implementation and maintenance.(3 hours)
- Tools for software development/computer-aided software engineering (CASE) including IDE's, SDK's and software version control systems. (3 hours)
- Object-Oriented design tools – UML, documentation standards (RAD, SDD, ODD, Test Plans and Reports). (3 hours)
- The software design process - Requirements, Analysis, System Design, Object Design, Implementation, Testing. (12 hours)
- Project teamwork, management and communication. (3 hours)
- Software system case studies. (3 hours)
- Large software design project (determined each semester). (15 hours)

### Course Objectives and Their Relationship to Program Objectives:

The objectives of this course are to expose students to formal processes for the design, implementation and management of large software systems. Students experience these processes through case studies and a large software design project through the entire semester. A student should understand (i) the object-oriented process from requirements through testing, (ii) the use of standard software engineering documentation for project development including RAD, SDD, ODD as well as documentation tools - UML, and (iii) the fundamentals of software project management. A student should be able to design efficient and complex object-oriented

software to solve complex engineering problems. A student should master modern tools for computer-aided software engineering (CASE) including IDE's, SDK, and version control systems. [Program Objectives this course addresses: 1, 2, and 4.]

### **Course Outcomes and Their Relationship to Program Outcomes:**

The following are the course outcomes and the subset of Program Outcomes (numbered 1-7 in square braces "[ ]") they address:

- Interact with a “client” to elicit project requirements by developing and refining scenarios and use cases. [1,2,3,4,5]
- Extract an Object Model and Dynamic Model of system functionality and performance from the requirements. [1,2,3,5]
- Design and implement structured, robust, maintainable object-oriented systems across multiple platforms and appropriate programming languages from the specifications developed. [1,2,6,7]
- Develop teamwork and management skills to divide tasks and effectively develop projects in large software teams. [3,5,7]
- Produce industry standard documentation from requirements, analysis, and design through testing and verification. [1,2,3].
- Research, evaluate and use various CASE tools for object-oriented software engineering including IDE's, SDK's, and version control systems(e.g. GIT). [1,2,6,7]

### **Contribution of Course to Meeting the Professional Component**

Engineering Topics: 100%

### **Computer Usage:**

Students use a wide variety of platforms, CASE tools, and programming environments throughout the semester as this is a software engineering course. The course also makes extensive use of Internet resources for research, communication, and implementation of systems and subsystems. The course has a web site, which has downloadable software and documents and reference links as well as makes extensive use of a courseware environment (Laulima) including a Wiki for collaborative document generation, discussion forums and chat rooms for communication, and file repository for shared code.

### **Design Credits and Features:**

EE 467 has 2 design credits. The course revolves around the complete design, from requirements to testing, of a large software system chosen each semester.

**Person(s) Preparing Syllabus and Date:** T. Dobry , February 14 2009