COMP 3000, Object-Oriented Programming for Engineers and Scientists

**Credit hours:** 3 lecture  
**Contact hours:** 3 lecture

**Catalog Description:** Fundamentals of object-oriented design and programming principles; data abstraction, identifying objects, problem decomposition, design and implementation of classes.

**Prerequisites:** Departmental approval  
**Corequisites:** None

**Required Course** (ELEC, INSY, WIRE)

**Instructor or Course Coordinator:** Dr. Alvin Lim

**Required Textbook**  

**Software Used**  
- GCC (https://gcc.gnu.org/)
- jGRASP IDE (http://jgrasp.org/)

**Course Outcomes**  
The student will be able to  
- Be able to apply an iterative, incremental software development process.  
- Be able to apply good software engineering principles.  
- Be able to apply concepts and techniques of object-oriented design and programming.  
- Be able to perform fundamental testing and debugging activities.  
- Demonstrate an understanding of objectives 1 through 4 by developing full project folders (containing requirements analysis, design, source code and testing sections) for C++ programming projects throughout the semester.  
- Understand and be able to explain how techniques from this class can be applied to the development of significantly larger computer programs.  
- Be able to discuss fundamental time and space implications of various design decisions and C++ programming constructs.

**Topics Covered**  
- Fundamental Computer Concepts: Hardware, Integrated Development Environment (3 hours)  
- Fundamental Programming Language Concepts (1.5 hours)  
- Software Engineering Principles, Spiral Software Development and "Extreme Programming" (3 hours)  
- Critical Differences between the "Real World" and the Computer Domain (1.5 hours)  
- Fundamental C++ Constructs: "Hello, World" through Control Constructs (3 hours)  
- Functions (4.5 hours)  
- Arrays (1.5 hours)
Object-Oriented Concepts and Benefits (1.5 hours)
Fundamental C++ Classes and Objects (6.0 hours)
Dynamic Memory Allocation and Pointers (4.5 hours)
Inheritance of Operations (1.5 hours)
Polymorphism (3 hours)
The Standard Template Library (3 hours)
Recursion (1.5 hours)
Exception Handling (3.0 hours)
Exams (3 hours)

Course Requirements
- Programming Assignments (50%)
- Midterm Exam (20%)
- Final Exam (30%)

Syllabus prepared: Spring 2016