東吳大學
SOOCHOW UNIVERSITY

**Academic Inquiries**: Soochow University
**Phone**: 886-2-2311-1531

## Course Information

| | |
|---|---|
| Course Title：<br>EGR225 – Computing for Engineering Simulation | Semester：Winter 2019 |
| **Class Hours:** Monday through Friday, 160 minutes each day | |
| **Discussion Section:** 3 hours each week | **Field Trip:** According to professors' teaching plan |

## Prerequisites

Students are expected to be proficient and able to code in one of the commonly-used programming languages such as C, Java or Matlab. Code examples discussed in the course will be in C. CS courses in Introduction to Algorithms and Discrete Mathematics or similar are highly recommended as a prerequisites.

## Textbooks and Reference

B. P. Zeigler, H. S. Sarjoughjan, R. Duboz, J.-C. Soulie, Guide to Modeling and Simulation of Systems of Systems (Simulation Foundations, Methods and Applications), Springer,  2012th Edition

## Course Outcomes

The expected outcomes of this course include

- Students will learn to simulate engineering systems and devices of small-to-moderate complexity using one of the commonly-used programming languages.
- Students will learn to apply generic methods and modeling techniques to specific engineering problems.
- Write computational code that complies with given specifications for input and output.
- Apply software engineering techniques to divide large progects into smaller parts, components and methods.
- Analyze and evaluate various algorithms for efficiency.
- Learn to test and validate a given algorithms or implementation of an algorithm based on the input/output specifications.
- Devise and data structures that most efficiently represent the modeled systems.
- Employ visualization techniques to render software more user-friendly and accessible.
- Understand the concept of software life-cycle and the best practices that ensure durability and extendibility of software.

## Course Description

The course is focused on techniques, algorithms and good practices of software development for engineering applications. Students are expected to have a working knowledge of a programming language commonly used in engineering such as C, Java or Matlab. Programming assignments and tests will be given. The important concepts covered by this course include construction of quantitative models that describe realistic systems and devices, designing simulation algorithms, decomposition of algorithms into reusable methods (abstraction) and validation.

## Weekly Schedule (Jan. 7 – 28, 2019)

| Week | Day | Topic |
|---|---|---|
| 1 | 1 | Interactive and compiled languages |
| | 2 | Computer resources |
| | 3 | Numerical accuracy<br>Numerical efficiency |
| | 4 | Software structure; methods and modules |
| | 5 | General approaches in software engineering |
| 2 | 1 | Intrinsic and derived data types and structures |
| | 2 | Procedural and object-oriented programming |
| | 3 | Procedural and object-oriented programming<br>Visualization |
| | 4 | Visualization |
| | 5 | **Midterm exam** |
| 3 | 1 | Testing and validation |
| | 2 | Testing and validation |
| | 3 | Software lifecycle and long-term development strategies |
| | 4 | Software lifecycle and long-term development strategies |
| | 5 | Practical examples |
| | Jan. 27 or 28 | **FINAL EXAM DAY** |

**Note**：

1. The course schedule is subject to change (please follow the announcements in class). If time permits, we will also cover miscellaneous topics based on students' interests.

2. The instructor will offer 3 hours of discussion or help session each week and students are encouraged to seek extra help by making arrangements with the instructor or the teaching assistant.

| **Grading** | | |
|---|---|---|
| Homework<br>(2 programming assignments, 20% each) | 40% | |
| Midterm Exam (Week 2) | 20% | |
| Final Exam: (Week 3) | 40% | |

Upon successful completion of this course, students will receive a final grade. A final grade is a numerical value that demonstrates a degree of understanding of the course materials. A final grade will be determined as follows:

| Grade | Percentage |
|---|---|
| A | 80-100 |
| B | 70-79 |
| C | 60-69 |
| D | 50-59 |
| E | 0-49 |

**Originality.** All assignments, including code, must be written or completed by you “from scratch”, and not copied from internet, books, or other sources.

**Cheating.** Cheating can be generally defined as conscious or unconscious appropriation (copying and passing for your own) of ideas, work, text, code or similar intellectual products of someone else without proper acknowledgment. Cheating is detrimental to the educational process. Confirmed instances of cheating will result in a reduced grade or further disciplinary measures. Cheating does not involve working in groups, discussing home assignments with other students, asking other students for help between classes, or finding answers to questions on the internet. However, direct copying of answers or assignments without acknowledgment is a form of cheating. The over-riding principle of academic integrity is complete intellectual honesty.