CSC444F:
Software Engineering I

Mou Hu
mou.hu@utoronto.ca

Main Topics of This Course

- **The software development process.**
- **Software requirements and specifications.**
- **Software design techniques.**
- **Techniques for developing large software systems.**
- **CASE tools and software development environments.**
- **Software testing, documentation and maintenance.**

(Prerequisite: ECE344S)

Course Objectives

- **Goal 1**

  to help students to develop skills that will enable them to construct software of high quality – software that is reliable, and that is reasonably easy to understand, modify and maintain

- **Goal 2**

  to foster an understanding of why these skills are important

Teaching Methods

- **Lectures (2 hours per week)**

  …are compulsory …cover core material

- **Tutorial and Practical Sessions (2.5 hours per week)**

  …are compulsory …are used for tutorials and for the team projects including team meetings

  You will need to meet with your team regularly

Sections

- **There are 3 lecture sections:**

  Sections 1, 2, and 3

  You have been assigned to one of them.

- **There are 6 practical sections:**

  you should pick 2 partners to form a group of 3. Each group can choose a preferred practical section. We will make the final decision to balance the number of each section. This will be done at this week's practical session.

Team Project (1/3)

- **Project will follow the waterfall model, which has 5 phases.**

  Phase 1: Requirement Engineering

  Phase 2: Design

  Phase 3: Implementation

  Phase 4: Testing

  Phase 5: Maintenance

- **We will give you initial requirements. Each team should follow the model to develop an operational software system**

Team Project (2/3)

- **Phase 1: Requirements Analysis (2 weeks)**
  Input: an initial requirements specification
  Output: a complete requirements specification
- **Phase 2: Software Design (3 weeks)**
  Input: requirements specification
  Output: design document
- **Phase 3: Software Implementation (coding, unit testing, integration, integration testing) (3 weeks)**
  Input: design document
  Output: program and test report

Team Project (3/3)

- **Phase 4: Software System Testing (1 week, overlapping with phase 3)**

  Input: requirements specification, design document, program

  Output: system test plan, system test report, and fully tested software
- **Phase 5: Software Maintenance (4 weeks)**
  Input: change request

  Output: new version of software, fully tested and well documented
- **Presentation**

Grading

- **Project**

  Phase 1: 4%

  Phase 2: 10%

  Phase 3: 10%

  Phase 4: 6%

  Phase 5: 10%

  Presentation: 10%
- **Final exam: 50%**

Books

- **Course Textbook (Required)**

  Hans van Vliet, "Software Engineering: Principles and Practice (Second Edition)" Wiley
- **References**

  To be posted

Lecture Schedule (1/2)

- Week 1 - No lecture
- Week 2 - Orientation and introduction
- Week 3 - Project management and software life cycle
- Week 4 - Requirement engineering
- Week 5, 6, 7 - Software modeling, design, and architecture

Lecture Schedule (2/2)

- Week 8, 9 - Software implementation
- Week 10, 11 - Software verification and validation
- Week 12 - Software maintenance and evolution
- Week 13 - Software quality
- Week 14 - Course review