

# Introduction and Overview

# Mission

1. Understand the professional responsibilities of software engineers.
2. Understand the role of precise specifications in software development.
3. Learn how to read and use specifications in program design, implementation, testing, and inspection.
4. Learn the basic principles of software design with emphasis on programs that are sequential, terminating, and composed of modules.

## Work Plan

- Lectures (see schedule in course outline)
- A practical exercise every week
  - Done individually
  - All programs will be written in C
- 2 midterm tests done in class
- Final exam

## Texts

1. **Required:** M. v. Mohrenschildt, *SE3K04 Lecture Notes*, available in bookstore (soon).
2. **Required:** Ghezzi, Jazayeri and Mandrioli *Fundamentals of Software Engineering*
3. **Optional:** F. P. Brooks, Jr., *The Mythical Man-Month, Anniversary Edition*, Addison Wesley, 1995.
4. **Optional:** D. M. Hoffman and D. M. Weiss, *Software Fundamentals: Collected Papers* by David L. Parnas, Addison Wesley, 2001.

# Mechanics

- Course web site:  
<http://www.cas.mcmaster.ca/~curette/SE3M04/2003/>
- Teaching assistants
  - Two EE graduate students
  - \* Mohammed Smadi and Feng (Stephen) Tao
  - Provide assistance with lab exercises and course material
  - Mark assignments
- Each student is required to keep a log

## Selected Policy Statements

1. I would appreciate your suggestions on how I can improve my teaching methods.
2. Regular class attendance is expected.
3. You are urged to ask questions during class.
4. You are welcome to discuss lab exercises with other students, but all such interactions must be recorded in your log.
5. Your final documentation and code must be your own.
6. Lab exercises may not be turned in late and midterm tests may not be taken later without **prior** approval from the instructor.

# Labs and Tutorials

- Lab - Virtual !
- Tutorials will be used to go over concepts useful for the assignment(s).

# What is "Software Engineering"

- Engineering is a discipline
- **Goal:** Creation of a Product
- The Product is specified, documented, and it is possible to verify that it performs as it is supposed to
- An engineer is responsible for his/her product (Not a disclaimer!)

# “Engineered” Software Products

- Pure software products
  - COTS: Microsoft Word, Matlab, Spice
  - Banking system, MUGSI
- Software as part of others products
  - (Car, Airplane, Nuclear Power Plant, X-ray-machine)
- Software used in the design of products
  - program computing the beams of a bridge

# Mission and Safety Critical software

Definition:

- failure causes harm to life or environment
- financial disaster (recalls)

Who of you ever wrote a program larger than N lines which worked perfectly the first time?

How can we trust software consisting of 1'000'000 lines of code or more?

## Goals of this course:

Learn how to develop (large) software products that

- are reliable
- really do what they are supposed to
- can be verified
- can be developed by a team
- can be easily maintained

# Engineering Principles

- Accept individual responsibility
  - Social
  - Professional
  - Ethical
  - Environmental
- Solve the Real Problem
- Be honest about the capabilities
- Produce Reviewable Designs (Documentation)
- Specify and document your software
- State the limitations of your software