SE 101

Introduction to Methods of Software Engineering

http://www.student.cs.uwaterloo.ca/~se101

Prof. Joanne (Jo) Atlee
DC 2337, x4871
jmatlee@se.uwaterloo.ca
Typical Software Practices

We can do better.
Agenda

- SE 101
- Professionalism
- Software Engineering
Course Personnel

Prof. Joanne (Jo) Atlee
DC 2337, x4871
jmatlee@se.uwaterloo.ca
Office Hours: Mondays 4:00-5:30
Thursdays 12:30-1:30
or by appointment

Benoit Laroche - SE 101/WEEF TA
E2 1318
blaroche@student.cs.uwaterloo.ca

Catherine Moore - CS 133 Tutor
ce3moore@student.cs.uwaterloo.ca

Julie Vale - SE 101 graduate TA
jrvale@engmail.uwaterloo.ca
**SE 101 Objectives**

SE 101 is an engineering **concepts course**  
Course weight  0.25

- **Software Engineering concepts**  
to give you a sense of what software engineering, as a discipline, is all about

- **Professional Engineering**  
to give you a sense of what engineering, as a profession, is about.

- **Technical writing, Grammar**  
to improve your writing skills
SE 101 Lectures

1. Intro to Software Engineering
2. Inexact quantities
3. Error propagation
4. Floating-point numbers
5. Design process
6. Teamwork
7. Project planning
8. Decision making
9. Professional Engineering
10. Software quality
11. Software safety
12. Intellectual property
SE 101 Lectures

1. Intro to Software Engineering
2. Inexact quantities
3. Error propagation
4. Floating-point numbers
5. Design process
6. Teamwork
7. Project planning
8. Decision making
9. Professional Engineering
10. Software quality
11. Software safety
12. Intellectual property

Numerical precision
SE 101 Lectures

1. Intro to Software Engineering
2. Inexact quantities
3. Error propagation
4. Floating-point numbers
5. Design process
6. Teamwork
7. Project planning
8. Decision making
9. Professional Engineering
10. Software quality
11. Software safety
12. Intellectual property

Numerical precision
Professionalism
SE 101 Lectures

1. Intro to Software Engineering
2. Inexact quantities
3. Error propagation
4. Floating-point numbers
5. Design process
6. Teamwork
7. Project planning
8. Decision making
9. Professional Engineering
10. Software quality
11. Software safety
12. Intellectual property

Numerical precision

Professionalism

Software Engineering
SE 101 Labs

• Design project / Design report (Lego robots)
• Quizzes
• Videos
• Ethics workshop
• WHMIS
Course Deliverables

- Web reviews (best 7 of 8) 10%
- In-lab quizzes (3) 30%
- Lab assignments 20%
  - Technical memo
  - Scheduling
  - Ethics memo
- Design Report 40%
Web Reviews

Web-based reviews cover lecture material and assigned readings:

• Review last week’s lecture material (if technical)

• Preview this week’s lecture topic (if nontechnical)
  
  You read about traditional engineering concepts, so that we can use lecture time to discuss how these concepts apply to software engineering and to case studies.

• Cover grammar
Web Reviews

First review due by Monday Sept. 20, at 5:00 p.m.

- **Introduction to Engineering** IPE Ch. 1
- **Punctuation** Dupré Intro, 15, 23, 29, 80, 93, 139

Web reviews will be accessible over the Internet, via a link from the course Web page.

You can work on the Web reviews in pairs, but everyone must submit a review to receive credit.

If you have technical difficulties, **let us know ASAP**. The reviews are untimed. But you must complete each review by Monday at 5:00 p.m., no exceptions.
SE 101 Textbooks


Agenda

• SE 101

• Professionalism

• Software Engineering
Expectations

We expect you to

• Be respectful of others in lecture/lab; do not arrive late or make noise

• Read assigned text before lecture/lab

• Be respectful of the course personnel

• Come see us if you have a problem or complaint
  ➔ don’t whine
  ➔ don’t stew
Expectations

You can expect us to

• Be accessible and approachable

• Be respectful of you and your opinions

• Involve you in any major change in the course deliverables or marking scheme

• Listen to any constructive comments you have about the course and the program
You are Responsible

You are enrolled in a professional program, and we expect you to **behave professionally**.

- Hand in for personal credit **only work done by you**; properly acknowledge any help you receive.
- Do not disrupt others in class by arriving late or making noise.
- Get a medical certificate **ahead of time**, if you are going to miss a quiz, assignment, or exam for health reasons.
Academic Dishonesty

**Cheating on exams** includes using unauthorized aids or communicating in any way with others during an examination.

The penalty for cheating on a test or exam will range from suspension for one or more terms to expulsion from the university.
Academic Dishonesty

Cheating on assignments and projects includes copying another student's solution and submitting it as your own, allowing another student to copy your solution, collaborating excessively with another student or decompiling sample solutions.

Individual assignments must be worked through by yourself and written in your own words. Working out solutions in detail with someone else and submitting the work as your own is a violation of the Student Academic Discipline Policy.
University Rules and Policies

• **University Calendar**
  
  http://www.ucalendar.uwaterloo.ca
  
  → Rules and policies specific to academic programs
  → Co-op process
  → Course descriptions

• **University Policies**
  
  http://www.adm.uwaterloo.ca/infosec/Policies/Policies_num.htm
  
  → Policy 71 - Student Academic Discipline Policy
  → Policy 70 - Student Grievance Policy
  → Policy 64 - Use of Proprietary Computer Software
  → others
You can Succeed

• Our admission standards are high: if you have been admitted, we believe that you have the **capability** to succeed in the program

• Attribution rates in Engineering are low compared to other institutions

• But you will need to **work** to **succeed**. Success will not come naturally, as it did in high school.
People to Know

- **Classmates** - study groups
- **Course TAs** - office hours
- **SE TAs** - office hours, often are around
- **Course professors** - office hours
- **SE program officers** - academic advising
- **Directors of First-Year Studies** - help sessions
- **Counselling Services** - study skills courses
Agenda

• SE 101

• Professionalism

• Software Engineering
Software Engineering

Software Engineering is a systematic and disciplined approach to developing software.

It applies both computer science and engineering principles and practices to the creation, operation and maintenance of software systems.
SE = CS + Engineering

Computer Science  +  Engineering

- data management
- data transformations
- designs patterns
- algorithm paradigms
- programming languages
- human-computer interfaces

- disciplined processes
- scalable design principles
- design evaluation
- effective documentation
- coordinated teams
- non-functional metrics
  (performance, reliability, maintainability, ease of use)
Evolution of an Engineering Discipline

- Virtuosos and talented amateurs
- Intuition and brute force
- Haphazard progress
- Casual transmission
- Extravagant use of available materials
- Manufacture for use rather than for sale

- Skilled craftsmen
- Established procedure
- Pragmatic refinement
- Training in mechanics
- Economic concern for cost and supply of materials
- Manufacture for sale

- Educated professionals
- Analysis and theory
- Progress relies on science
- Educated professional class
- Enabling new applications through analysis
- Market segmentation by product variety

Example: Civil Engineering

1700: statics
1700: strength of materials

1750: properties of materials
1850: full analysis of bridge

100 A.D.: Romans

Example: Chemical Engineering

1800: atomic theory

Science

Production

Professional Engineering

Commercial

1890-1915: unit operations

1790: first industrial alkali process

Emergence of Software Engineering

1960s: data structures, algorithms
1990s: design patterns, architecture patterns

1980s: software development processes

Science

Production

Professional Engineering

Craft

Commercial

Space shuttle software
Aeronautical software
Defense software
Compiler construction

Summary

Software engineering is an emerging *engineering discipline* — despite the state of the current practice of software development.

- Systematic development processes
- Analysis and theory of designs
- Design patterns for recurring problems
- Professional societies, code of ethics
Readings

• For today’s lecture:
  • IPE Ch. 1- An Introduction to Engineering

• For next week’s lecture:
  • IPE Ch. 10, 11 - Inexact quantities

• Web review due Monday Sept. 20, 5:00 p.m. on
  • IPE Ch. 1- An Introduction to Engineering
  • Dupré: Intro, 15, 23, 29, 80, 93, 139