

COMPSCI 1JC3
Introduction to Computational Thinking
Fall 2017

00 Introduction to the Course

William M. Farmer

Department of Computing and Software
Faculty of Engineering
McMaster University

September 6, 2017



Outline

00 Introduction to the Course (today).

1. Instructional staff.
2. Student composition.
3. My goals.
4. What computing is.
5. What computational thinking is.
6. Mission of this course.
7. Required resources.
8. Work plan.
9. Marking scheme.
10. Other resources and policy issues.

01 What is Computational Thinking? (tomorrow).

02 What is Functional Programming? (next Monday).

Instructor: Dr. William M. Farmer

- Professor, Dept. of Computing and Software.
- B.A., University of Notre Dame.
M.A., M.S., Ph.D., University of Wisconsin-Madison.
- P.Eng. (Licensed Professional Engineer in Ontario).
- **Industrial experience.** Research computer scientist for 12 years at The MITRE Corporation in Bedford, MA, USA.
- **Teaching:** Discrete mathematics, logic, principles of programming, software design, communication skills.
- **Research:** Applied logic, mechanized mathematics, formal methods, mathematical knowledge management.
- **Software development:** In the early 1990s, I developed the IMPS interactive theorem prover with J. D. Guttman, and F. J. Thayer at the MITRE Corporation.

My Contact Information

- Office: ITB 163.
- Email: wmfarmer@mcmaster.ca.
- Web: <http://imps.mcmaster.ca/wmfarmer/>.
- Office hours: To see me please send me a note with times.

Please use McMaster e-mail for all correspondence, NOT Avenue email.

Teaching Assistants

- Curtis d'Alves (dalvescb@mcmaster.ca) [Grad TA]
- Victor Chen (chenv5@mcmaster.ca) [Undergrad TA]
- Natalie Chin (chinnh@mcmaster.ca) [Undergrad TA]

Students

- Composition:
 - C01: 47 Computer Science students (33%).
 - C02: 95 Non Computer Science students (67%).
- I want to find out who you are and what are your goals!

Student Programs (iClicker)

What program are you in?

- A. Computer Science.
- B. Mathematics.
- C. Another Science program.
- D. Another Non-Science program.

My Goals

- I want to:
 1. Help you lay a foundation on which to build a sophisticated understanding of **computing**.
 2. Teach you how to **think like a computer scientist**.
 3. Provide you with opportunities for **active learning**.
- Will this course be challenging?
 1. Yes, it will take you out of your comfort zone.
 2. You will be expected to act as a computing professional.
 3. You are capable of meeting the challenge.

What is Computing?

- Computing is the development and use of **computer hardware and software** to:
 - ▶ **Solve** problems.
 - ▶ **Manage** information.
 - ▶ **Create** smart products.
 - ▶ **Explore** our world.
 - ▶ **Connect** to other people.
- Computing is a mixture of:
 - ▶ **Mathematics.**
 - ▶ **Science.**
 - ▶ **Engineering.**
 - ▶ **Technology.**
 - ▶ **Design.**
- **Computing is everywhere in today's world!**

What are the Fields in Computing?

- Computing includes:
 - ▶ **Computer Science**
(better name: **Computing Science and Engineering**).
 - ▶ **Computer Engineering.**
 - ▶ **Software Engineering.**
 - ▶ **Information Systems.**
 - ▶ **Information Technology.**
- These fields heavily overlap with each other.

What is Computational Thinking?

- **Computational thinking (CT)** is thinking inspired, supported, or enabled by computing.
- CT is thinking like a computer scientist.
- It is a skill that should be a part of everyone's education.

Mission

- The **mission of the course** is to:
 1. Introduce students to **computational thinking**.
 2. Help students begin the process of developing a sophisticated understanding of **computing**.
- **These goals are best achieved by active learning!**
 1. Engaging in interactive discussion.
 2. Doing hands-on programming.

Learning Objectives (Knowledge)

Students should know and understand:

1. What computational thinking is.
2. The key ideas, methods, technology, and areas of computing.
3. Functional programming language concepts.

Learning Objectives (Skills)

Students should be able to:

1. Design and write a simple functional program that satisfies a simple requirements specification.
2. Formulate a test plan for a simple functional program.
3. Analyze a simple functional program.
4. Use the Haskell Platform as a problem-solving tool.

By the end of the course, each student should be comfortable programming in Haskell.

Required Resources

- **Textbook 1:** Paul S. Wang, *From Computing to Computational Thinking* [abbreviated CCT], Chapman and Hall/CRC, 2015. ISBN-13: 978-1482217650. Available in a less expensive eBook edition.
 - ▶ The material in the entire text will be covered.
- **Textbook 2:** Simon Thompson, *Haskell: The Craft of Functional Programming (3rd Edition)* [abbreviated HCFP], Addison-Wesley, 2011. ISBN-13: 978-0201882957.
 - ▶ The material in Chapters 1–11 will be covered.
- **Equipment:** An iClicker remote.
- **Course web site:** All course materials will be available on Avenue to Learn at <http://avenue.mcmaster.ca/>.

Work Plan per Week

- **Preparation outside of class:** Study and programming.
 - ▶ 2-hours reading and studying CTT.
 - ▶ 2-hours reading HCFP and programming in Haskell.
- **Lecture:** High-level overview of course material.
 - ▶ 1-hour lecture on Monday given by the Instructor.
- **Discussion sessions:** Interactive discussion.
 - ▶ 1-hour discussion sessions on Wednesday and Thursday lead by the Instructor on material in CTT.
- **Tutorial:** Programming practice in Haskell.
 - ▶ 2-hour tutorial lead by the TAs.
- **Assessment:** What has been learned.
 - ▶ Five programming assignments.
 - ▶ Two midterm tests (on Fri, Oct 20, and Fri, Nov 17).
 - ▶ Final exam.

Class Participation

- Clicker questions.
 - ▶ Students will answer iClicker questions in the lectures, discussion sessions, and tutorials.
- Instructor's questions.
 - ▶ Students selected from a randomized list will be asked questions about the weekly reading assignment in CTT.
- Meaningfuls and memorables (M&Ms).
 - ▶ Students will submit three M&Ms at the end of each week to a discussion forum on Avenue.

Programming Experience (iClicker)

What is the level of your programming experience?

- A. Not sure what a computer program is.
- B. Have never written a computer program.
- C. Have written small programs.
- D. Have written large programs (100s of lines of code).
- E. Have been paid to do programming.

Programming Languages (iClicker)

In which of the following programming languages are you most proficient?

- A. Java or C#.
- B. C or C++.
- C. Python.
- D. Haskell.
- E. I am not (yet) proficient in programming.

Marking Scheme

Class participation	
a. Clicker questions	5%
b. Instructor's questions	5%
c. Meaningfuls and memorables (M&Ms)	5%
Programming assignments (5)	15%
Midterm test 1	15%
Midterm test 2	15%
Final exam	40%
Total	100%
Course review session bonus	2%

Other Resources and Policy Issues

1. Course evaluation.
2. Drop-In Centre and ETB 126.
3. iClickers.
4. Academic dishonesty.
5. Discrimination.
6. Academic accommodation.
7. Missed work.
8. Course modifications.
9. Cell phones are not allowed during lectures, discussion sessions, and tutorials.
10. No electronic devices may be used during exams.
11. Use McMaster email for all correspondence, NOT Avenue.