Computational Thinking

CPSC 100
Jessica Wong

Course website:
http://www.ugrad.cs.ubc.ca/~cs100

(If you go there now, you can find a preliminary copy of the slides for today)
Administrative Notes

• No lab this week
• Office hours start next next week
• Slides will be on the the website
  • Preliminary ones will be there before lecture
  • Final “inked” ones will be there after lecture
Learning Goals for Today
Meta point: what is a learning goal?

- Meta-meta point: what is a meta point?
  - Meta: Pertaining to a level above or beyond. For example, metadata is data that describes data, metalanguage is language that describes language, etc. (wiktionary)

- A learning goal is designed to tell you what information you should learn about a topic

- At the beginning of the topic, the learning goal may not even make any sense

- By the end of a topic, it should help you know what to study
Learning Goals for Today

• Understand what is computational thinking (CT)
• Become acquainted with some CT building blocks (namely, algorithm, abstraction, decomposition) by developing a sorting algorithm
Learning Goals for the Course

• **[CT Building Blocks]** use abstraction and decomposition to clarify and simplify the critical pieces of a problem; choose appropriate models and representations to aid in solving the problem; recognize potential shortcomings in a model or solution; explain at a high level how the computer works;

• **[CT Application]** explain specific algorithms that computers use to create applications of computational thinking (e.g., data mining), and

• **[CT Impact]** provide and explain examples of how computers impact their lives and what this means for them, both in terms of how the computer can add to their lives and in things that they have to be careful of, such as privacy and security
So how are you going learn?

- The best way to learn is by doing, so you’ll be doing a lot of “doing” in this course
- “Lectures” will involve:
  - Doing exercises
  - Discussing the readings (in small and large groups)
  - My explaining concepts
- Labs are time for you to
  - Get hands on practice with a computer
  - Work on your project
- You’ll have a group project (of your own choosing)
- You’ll want to do some exercises
What does this mean for you?

- You need to do the reading ahead of time
  - There will be a simple quiz for each reading – the quiz is to motivate you read, not to trip you up
- You need to come to class prepared to do things
  - Bring paper and a writing implement
  - Bring an iClicker (Note: you’re welcome to try iClicker Mobile/REEF, but it hasn’t worked well for past students)
  - If you have a laptop computer, you’ll probably want to bring it – but no goofing off
No, really, I mean it: no goofing off on your computer/phone/tablet in class.

• On the one hand, if you don’t want to learn and you only hurt yourself, that’s your problem.
• On the other hand, you’re not hurting only yourself.
• Researchers did a set of experiments where they asked students to multitask:
  • “Students … who were asked to multitask averaged 11 percent lower on their quiz.”
  • “Students … who were surrounded by laptops scored 17 percent lower on their quiz.”

What will we study about computational thinking (CT) in this course?

• **CT building blocks**: e.g., specifying algorithms
• **CT applications**: e.g., data mining, computer animation
• **CT impact**: e.g., how are jobs changing and how is that impacting society?
Abstraction is a key CT building block

“The most important and high-level thought process in computational thinking is the abstraction process. Abstraction is used in defining patterns, generalizing from instances, and parameterization. It is used to let one object stand for many.” – Jeanette Wing

http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf
What will we study about computational thinking (CT) in this course?

- **CT building blocks:** e.g., specifying algorithms
- **CT applications:** e.g., data mining, computer animation
- **CT impact:** e.g., how are jobs changing and how is that impacting society?
CT application: Computer animation

- What are the building blocks?
- How is it impacting the movie industry?

Tyrus Wong illustration for Bambi, 1942

Fluid Simulation for Computer Graphics

Robert Bridson
What will we study about computational thinking (CT) in this course?

• **CT building blocks**: e.g., specifying algorithms
• **CT applications**: e.g., data mining, computer animation
• **CT impact**: e.g., how are jobs changing and how is that impacting society?
CT impact

- Jobs today increasingly use computers (http://www.npr.org/blogs/money/2012/01/13/145039131/the-transformation-of-american-factory-jobs-in-one-company)
  - Designing cars
  - Factory workers
  - DJs
CT example: Human health

- **Building blocks**: DNA is digital!
- **Applications**: Genome databases
- **Impact**: Understanding rare genetic disorders
- **Impact**: Personal health care data is vulnerable
Note: this course is NOT a programming course

We’ll only do a very, very small amount of programming in a visual language called Snap!

If you want a course with substantial programming, consider CPSC 103 or CPSC 110.
How is your grade determined?

- Active class participation (5%)
- Reading quizzes (5%)
- Laboratory assignments (10%)
- Student-directed group project (15%)
- Computing in the news (5%)
- Midterm exams (10% each) – Feb 1 & Mar 13
- Final exam (40%) – date TBD
- Bonus 1% for filling out the class surveys: https://survey.ubc.ca/s/cpsc-100-student-experience-pre-2017w2/

Details are on the website. We’ll talk more about all things in lecture and in lab.
Where to find information

- Start with the course website (http://www.ugrad.cs.ubc.ca/~cs100); it has links to everything else you need
- This includes a link to Canvas – not Connect – which includes the course bulletin board
  - You are responsible for checking in at least once every day or two.
  - Please direct all course-material questions to the discussion threads, **not** to email – **you will get help much more quickly that way** – both from course staff and from other students.
Todos

- Reminder: reading quiz due Monday night (see Canvas)
- Make sure to follow up on To-Do’s listed on the web page and read announcements
- Check the web page to learn about course staff, labs, exams and more
- Take the survey and earn bonus 1%:
  https://survey.ubc.ca/s/cpsc-100-student-experience-pre-2017w2/

Acknowledgements: Thanks especially to Anne Condon and also to Meghan Allen, Warren Code, Jessica Dawson, Will Evans, Holger Hoos, Piam Kiarostami, Hassan Khosravi, Joseph Luk, Tristram Southey, Dave Tompkins, Kimberly Voll, Steve Wolfman, Jessica Wong and many others for valuable contributions to the slides and materials throughout.
Learning Goals for Today [review]

• Understand what is computational thinking (CT)
• Become acquainted with some CT building blocks (namely, algorithm, abstraction, decomposition) by developing a sorting algorithm