Course Information
How Computing Happens
Lecture 0

Some slides borrowed from Kurt Eiselt and Steve Wolfman.
Learning goal material from Beth Simon.

Rule #1: Ask questions!

Who I Am

Alan J. Hu (You can call me Alan or Prof. Hu.)
ajh@cs.ubc.ca
ICICS 325
office hours:
I will hang out after class (both sections)
if people have questions (MWF around 4pm)
(TTh around 5pm).
Feel free to stop by my office at other times,
but I may be busy or not there.

Other Introductions...

Benjamin Yu, UBC
Carl Weiman Science Education Fellow
Beth Simon, UC San Diego
Paul Denny, U of Auckland

What This Course Is About

Calendar description: Basic programming constructs,
data types, classes, interfaces, protocols and the design
of programs as interacting software components.
Prerequisites: Mathematics 12.

Who this course is for...

Although it is expected that you will have used a
computer prior to taking the course and that you are
familiar with basic keyboard and mouse operations, no
prior programming experience is assumed. This course
will teach you basic programming constructs that will
allow you to unleash your creativity and develop your own
applications software.
...but note this

If you have already received credit for CPSC124 and CPSC126 or for CPSC122 and CPSC128 then you cannot receive credit for CPSC111. If you have taken CPSC124 but not CPSC126 then you must take CPSC111 in order to advance through the computer science program. If you have taken CPSC122 but not CPSC128, please consult with a department advisor.

Administrative Stuff

CPSC 111 has two sections taught in parallel:

- Same labs, tutorials, homeworks, and exams.

You should attend the lecture section that works for you. (Or more than one if you wish!)

Labs and tutorials start on Monday; more about these soon…

(Do Lab #0 on your own this week.)

Administrative Stuff

Your textbook is Big Java (3rd Ed). (The 2nd edition is OK, too.)

You should get a copy. Seriously.

Read Chapter 1.

Thinking about Thinking:
What do I need to know from today’s lecture?

- Studying computing is a new experience for you
  - Always know what you are supposed to be learning otherwise you might miss it!

- Learning Goals
  - Each class will start with a slide of learning goals for the days lecture
  - Help you identify what you should be “getting out” of lecture for the day

- ASK
  - If you don’t understand those as we go through lecture!
Learning Goals
By the end of class today you will be able to...

- Find out more detailed information about the course on-line.
- Say the name of your textbook.
- Find the CS Learning Center.
- Describe the learning goals for the entire course.
- Explain why computer science is the most exciting thing to happen to humanity since writing.
- Explain, at a "cocktail party" level, how a transistor works, how a computer "computes"...

Course Learning Goals

- Be able to write simple (up to about a thousand lines long) programs that correctly express your intentions, in the Java programming language.
- Be able to read and understand the basic functionality of programs written in Java or similar programming languages.
- Change the way you think about computers!

This is a first course in computer science...
...but what is computer science?

"Computer science is as much about computers as astronomy is about telescopes.”
Edsger Dijkstra

This is a first course in computer science...
...but what is computer science?

"Computer science is how to harness the physical world to aid or exhibit 'thinking'.”
--Alan J. Hu

This is the most profound revolution in human history since the advent of writing!

The Most Profound Revolution...?

- Moveable Type Printing? Just lets you make more copies of some writing, more easily
- Steam Engine? Internal Combustion Engine? Just more efficient ways to harness physical objects to make us stronger.
- Boats, Trains, Planes? Just ways to move stuff somewhat faster and easier.
- Telephone, Radio, TV? Just ways to transmit information more easily.
Profound Revolutions
- Rocks, Sticks, Wheels, Levers: The first time humanity managed to use physical artifacts to augment our physical capabilities.
- Writing: The first time humanity managed to use physical artifacts to augment and encapsulate our memory.
- Computer Science: The first time humanity managed to use physical artifacts to augment and encapsulate thinking.

Harnessing the physical world to help us think...
- How to get create things that have “computational” behavior
  - Not the focus of this course
  - Technology dependent: sticks, gears, relays, vacuum tubes, transistors, DNA,…
- How to control that behavior to do interesting things.

Harnessing the physical world to help us think...
- Really simple example: slide rules
- Physical lengths add/subtract.
- What’s something that lets you multiply/divide via addition and subtraction?

Harnessing the physical world to help us think...
- Another example: a toilet valve
- This is a physical device that makes a decision!
- (In the 19th century, when the industrial revolution was going strong, a few people actually imagined building something like a modern computer out of steam-powered valves…)

Harnessing the physical world to help us think...
- Another example: a relay
- This makes a similar “decision” as a toilet valve.
- The first modern computers were built in the 1940s with relays…

Harnessing the physical world to help us think...
- Current technology: the MOS transistor
- This makes a similar “decision” as a toilet valve or a relay…
- Pretty much every computer or computer-like device is currently relying on MOS transistors…
With no charge carriers in depletion region, electricity can’t flow from source to drain.

Inversion

A high voltage gives gate a positive charge...

... which attracts electrons to other side...

... which makes the p-type material act like n-type...

... which lets electricity flow from source to drain!
Questions?