COMP 102: Computers and Computing
Lecture 1: Introduction

Instructor: Joelle Pineau (jpineau@cs.mcgill.ca)

Class web page: www.cs.mcgill.ca/~jpineau/comp102

Outline for today

• What are computers? What is computing?
• What will this course contain?
• Interesting success stories from the field of computer science.
• A brief history of computers and computing.
About You (as of Aug.29 2011)

- 20 Freshman Program
- 7 (Micro)-Biology
- 5 Education
- 3 Economics
- 3 Undeclared
- 2 Psychology
- 1 Music
- 1 Art history
- 1 Physics
- 1 Computer science
- 1 Kinesiology
- 1 Physiology
- 1 Environment

About Me

- Academic background:
  - B.A.Sc. in Engineering (U.Waterloo) 1993-1998
  - Prof at McGill 2004-

- Other teaching activities:
  - Artificial Intelligence (COMP-424), Planning Algorithms (COMP-765)

- Research interests:
  - Robotics
  - Artificial Intelligence
  - Machine learning
  - Personalized medicine and health-care
SmartWheeler project

Adaptive treatment design

**Background:** Deep brain stimulation is a new treatment for epilepsy.

**Problem:** Existing devices do not adapt to the patient’s condition.

**Idea:** Create an improved class of devices with adaptive control.
Am I taking the right course?

• I want to learn how to program.

• I want to know how to use Word and Excel.

• I want to build intelligent robots.

• I want to design a cooler Web page.

Goals for the course

• Discuss interesting facts and ideas related to computers and computing.
  – How computers represent data.
  – How we can give instructions to computers.
  – What problems computers can (or not) solve.

• Develop your Computational Thinking
  • What is Computational Thinking? Ability to solve problems logically, algorithmically, recursively, efficiently, scientifically, innovatively.

• Explore modern applications of computer science.
Course Syllabus

Part 1: What do we know about computers and computation?
- Storing information (bits, bytes, memory, databases)
- Manipulating data (searching, sorting and more)
- Hiding data (how and why)
- Examining what problems can be solved and how fast.

Part 2: What can we do with computers and computation?
- Designing simulations and games
- Building robots and artificial intelligence
- Solving problems in biology
- Music, images and graphics

Evaluation method

- Weekly assignments (40%)
  - Individual work.
  - Some theory, some application.
  - No formal programming required.
  - Late assignments will NOT be accepted (no exception). But only the marks from the 8 best assignments (out of 11) will count.
- One midterm examination, in class (20%).
- One final examination (40%).
Other details about the course

- **Textbook:** None required.
  
  Suggested reading: *The Information: A History, a Theory, a Flood* by James Gleick
  
  (Available in most local and online bookstores.)

- **Office hours:**
  - Joelle Pineau: McConnell 106N. Tues. 1-2pm.

- Course material and announcements available on class website:
  

---

What is this?

The Antikythera Mechanism, built by the Greeks some time between 140 and 100 B.C., predicted solar eclipses and organized the calendar in the four-year cycles of the Olympiad, forerunner of the modern Olympic Games.

*(Antikythera Mechanism Research Project)*
Back when *computers* were people

The ancestors of the computer

- Counting machines (abacus, etc.)
- Analog computers to perform astronomical calculations.
- Intricate mechanical automatons.

Why aren’t these computers? What’s missing?
Jacquard’s innovation

• In 1801, Joseph Marie Jacquard invented a machine that could weave a specific pattern based on information automatically read from punched wooden cards.

Charles Babbage (1791-1871)

• Lived in England.
• He was a polymath (solved problems from Astronomy to Zoology, in particular mathematics, philosophy, engineering).
• Travelled across England, studying looms, forges, glassworks.
• Invented:
  – The cowcatcher,
  – Flat-rate postage,
  – Operations research
• Held the Lucasian Professorship of Mathematics at Cambridge (formerly Newton’s chair).
• His design for the analytical engine presaged much of the design of modern digital computers.
The Difference Engine

- Originally a **computer** was a job description of a person who created numerical tables, e.g. $\log_2(x)$
- Babbage had a scheme to **automate** table creation using **finite differences**.
  - The **Difference Engine No.1** required 25,000 parts, 15 tons, 8 feet high, and was never built.
  - His redesigned **Difference Engine No.2** was built in 1990 and actually worked!
Difference Engine: Main idea

- Let's play a game:
- 0, 1, 2, 3, 4, …?
- 3, 5, 7, 9, 11, …?
- 1, 4, 9, 16, 25, …?
- 1, 3, 6, 10, 15, ….?

Generating Even Numbers

\[
\begin{align*}
2 & \rightarrow 4 \rightarrow 6 \rightarrow 8 \rightarrow 10 \rightarrow 12 \\
+2 & \quad +2 \quad +2 \quad +2 \quad +2
\end{align*}
\]

- Instructions:
  1. Start with 2.
  2. To get the next number, add 2 to the previous one.
Generating Squares

1 → 4 → 9 → 16 → 25 → 36
3 → 5 → 7 → 9 → 1
+2 +2 +2 +2

- Instructions:
  1. Start with 1.
  2. Start the increment with 3.
  3. To get the next square, add the increment.
  4. To get the next increment, add 2 to the previous increment.

Naming the sequences

- The Evens: "2 2"

2 → 4 → 6 → 8 → 10 → 12
+2 +2 +2 +2 +2

- The Squares: "1 3 2"

1 → 4 → 9 → 16 → 25 → 36
3 → 5 → 7 → 9 → 1
+2 +2 +2 +2
Programming the Different Engine

• To produce the Evens, we enter "2 2" into the Difference Engine and turn the crank.
• For the Squares, enter "1 3 2"
• For the Triangle numbers, enter "1 2 1"

Why is this “programming”?

• There’s a way to say what you want to say, in a way the machine understands.
• Simple operations (in this case, repeated addition) used to build up more complex objects.
Limitations of the Difference Engine

- It was engineered for a very specific task.
  - It can only compute polynomial functions.
  - Some other functions (e.g. log, trig functions) can be approximated by polynomials.
  - To perform very different calculations, we would need to re-build the machine.

The Analytical Engine

- Much more general, with many of the characteristics of a modern-day computers.
  - Input: program and data can be entered through punched cards.
  - Output: through printer, curve plotter and bell.
  - Internal memory.
  - Arithmetical unit, capable of basic mathematical operations.

- Introduced the notion of a conditional statement.
  - “If you observe condition X, apply operation Y.”

- Never built, more of a thought experiment.
The first programmer: Ada Lovelace (1815-1852)

- Ada August, Countess of Lovelace
  - Daughter of the writer Lord Byron.

- Designed many sequences of Babbage’s machine’s basic operations.

- Was the first to see it as an *engine of information* (not just an engine of numbers).

- Used a number of modern software design elements, e.g. the loop, the variable.

  *Loop* = Repeat an operation until a specified condition is met.

  *Variable* = A receptacle, capable of storing a number.

1940: First demo of remote computing

- The Complex Number Calculator.

- Designed by Bell Telephone Labs.

- On its own, just a fancy calculator.

- Noteworthy because calculations were performed remotely

  (NYC - Dartmouth) using a Teletype connected via phone lines.
1944: Room-sized computers

- Harvard’s Mark-1.
- 765,000 component parts:
  - switches, relays, rotating shafts, clutches
  - hundreds of miles of wire
- Used to produce mathematical tables.
- Programmable by punched-paper tape.

1948: ENIAC

- The Electronic Numerical Integrator And Computer (ENIAC)
- First purely electronic digital computer.
- Built at University of Pennsylvania to calculate artillery firing tables for the US Army.
- 80 feet wide, 27 tons, 120,000 pieces.
- 300 times faster than the Mark 1 at addition.
- Programmable by rewiring
  (switches and cables).
- No memory.
1945: von Neumann’s stored-program computer

- Early computers had fixed programs.
- Key idea is to hold the program in memory as a series of instructions.
- The control unit (i.e. “computer”) treats these instructions in the same way as other data.
- Allows the program to change (even as it is running.)

The Transistor Revolution

- Early days: computers worked with vacuum tubes.
  - Think of this as an “on/off” switch.
  - Switch is caused by electrons.
  - So we call this “electronics”.
  - Vacuum tubes were about size of a modern light bulb.
  - ENIAC had 17,000 vacuum tubes.

- 1947: engineers at Bell Laboratories invent the transistor.
  - Also an electronic on/off switch.
  - Much smaller, less hot, less expensive.

- Today:
  - Intel’s recent processors: 781 million transistors.
Integrated Circuits

- Integration of large numbers of tiny transistors into a small chip
- Mass production capability, reliability, and building-block approach to circuit design ensured the rapid adoption of standardized ICs

Personal Computing

- By late 1970’s, price of computer processors was much more affordable. Companies were starting to produce “microcomputers”.
  - But not clear whether there was a market for personal computers.
  - Computers were still seen as “number-crunchers”.
- In early 80’s, hobbyists starting buying machines like Apple II and IBM PC.
- Two key developments:
  - Graphical User Interfaces (“GUI”)
  - Application software
What came next?

- **1946**: “At most six electronic digital computers should be sufficient to satisfy the computing needs of the entire United States.” Howard Aiken, principal designer of the Mark I.

- **1977**: “There is no reason for any individual to have a computer in his home.” Ken Olsen, CEO of Digital Equipment Corporation

- **2011**: How many computers have you used today?
Computers and your health

www.wisc.edu

www.bcm.com

www.davincisurgery.com

Computers and your social life

facebook

YouTube

myspace.com

Joelle Pineau
Let’s start with a few basic concepts

- What is **computing**?
  - Take an **input**
    - e.g. text, numbers, image, sound
  - **Calculate** for a while
    - e.g. add/subtract, filter/amplify, summarize
  - Produce an **output**
    - e.g. new text, new numbers, error message
  - **Store** the results

- The **computer** is the machine which allows us to do all of this.

*We begin the course by trying to understand these different components.*
Something to think about

- Define computer science’s contribution to the world in one-word:
  REDUCTION
  - Computer scientists solve problems by reducing them to simpler problems.
  - Use simple operations to build up more complex objects.

"We are trying to build a machine to do all kinds of different things simply by programming rather than by the addition of extra apparatus," Turing, 1947.

We will see this concept arise in a number of instances throughout the semester.

Take-home message

- You should understand:
  - how the difference engine works.
  - the link between what we can compute
    - polynomials vs full weather prediction
  and how we compute it
    - difference engine vs modern computer with millions of transistors

- For the rest of the semester, we explore the current state of computing.
  - Minor emphasis on hardware&engineering, more emphasis on the science behind computer science.

- Coming up next:
  - What is data, and how is it represented?