McGill
School of Computer Science

# COMP 102 : Computers and Computing

*Fall 2016, McGill University*

## Syllabus + Links

**Lecture 1**. ***Due diligence*** *for the course +* ***Overview****.*

- Hour of Code
- Computer Science Unplugged
- This Is MEGA-Mathematics!

**Lecture 2. *A short history*.** Pre-electronic computation. First computing devices. Development of underlying concepts to computing.

- Antikythera Mechanism (youtube) (wiring diagram of gears)
- Napier's bones (simulator)
- Jacquard Loom
- Babbage's machine (Video series episode 1 through 4)
- History of Computers
- History of Computer Science
  - from J Shallit (Waterloo, CS 134)
  - from J Kopplin
- The Enigma, a biography of A Turing (The Imitation Game, movie)
- Turing Award Winners

**Lecture 3. *Modern Information Technology.***

- ***Workplace transformations***: Google Drive + Forms + Doodle + Trello + HipChat.
- ***Multidimensional on-line personalities*:** from websites to integrated social media and mobile computing.
- ***Internet of Things:*** IFTTT and NEST technologies. (wiki)

**Lecture 4. *Fundamentals*.** How computer science often uses simple, abstract, elegant models that allow us to see the main issues: the Turing Machine, a surprisingly powerful model of computing.

- Definition of a Turing machine (Cambridge University)
- Turing machine simulator

**Lecture 5. *What "discrete" means to a computer scientist*.** Bits, circuits, logic.

**Lecture 6. *The Central Dogma of Computer Science*:** Input -> Algorithms -> Output. And **how this is made concrete:** programs, variables, assignment, operations.

- Code: Blocks and JavaScript
- Interactive Java Interpreter

**Lecture 7. *Instructing computers to react to their environment*:** Conditional execution.

- Stackoverflow

**Lecture 8. *Enslaving computers*.** Iteration and looping combined with conditionals.

**Lecture 9. *Popular media and Computer Science*** (A break from geeky programming…) Life lessons for seducing computer scientists: memorizing important moments from 2001 A Space Odyssey, Wargames, Ghost in the Shell, Mr Robot, and HBO's Silicon Valley.

- Bachmanity

**Lecture 10. *Compression: The Lost Art*.** From silly to serious algorithms.

- Pied Piper Inc.
- Compression
- CS Unplugged – Compression
- Scratch
- Scratch version of exercises

**Lecture 11. *Efficiently repaving Montreal streets.*** Minimum spanning trees.

- This Is MEGA-Mathematics!

**Lecture 12. *Computational Complexity.*** Coloring maps with two colors versus coloring maps with three (or more colors). Tractable versus intractable problems.

- This Is MEGA-Mathematics!

**Lecture 13. *Syntactic sugar*** Functions, Modules, Packages. Software Engineering. ***Open Access + Open Source.***

- Size of Software Systems
- GIT
- Github
- Google Code

**Lecture 14. *Web and mobile applications.*** Building a smart phone app for voting

- Code
- App Lab

**Lecture 15. Cloud services and *Cloud Computing*.**

- Silicon Valley, s03.e9 minute: 5:50
- Google Cloud Computing.
- Amazon Web Services.

**Lecture 16. *Databases.*** Building them and querying them. SQL.

- SQL Fiddle

**Lecture 17.** ***How to train your computer.*** Determining the next song you will listen too. APIs and Spotify.

- Spotify
- Spotify API

**Lecture 18.** ***The Singularity and what it means for you****.* Neural networks and deep learning.

- Where will you be in 2040?
- Book: The singularity is near; R Kurzwell
- Ted talk: A university for the coming singularity
- Stephen Hawking in The Independent. We aren't taking artificial intelligence seriously enough.
- Centre for the Study of Existential Risk, Cambridge University.
- The dreams from Google's Artificial Intelligence

**Lecture 19.** ***Social Networks.*** Graph and network representation. Information flow.

**Lecture 20.** ***Privacy + Cryptography + Security + Hacking.*** Building our own cryptographic systems. Anonymous.org. NSA. Conspiracy theories. Chaos.

- Introduction to Hacking
- The Peruvian Coin Flip
- Public Key Encryption

**Lecture 21.** ***Robotics.*** An excuse for us all to play with Lego Mindstorm.

- Introduction to Mindstorm and programming software.

**Lecture 22.** ***Bioinformatics.*** Using computers to understand noisy, imprecise, illogical, irreproducible,dirty, non-discrete, and generally badly behaving biological systems like bacteria, fungi and humans.