COMP-202

Introduction to Computing 1

Section 1

Ernesto Posse

ENGMC 304

MWF 11:30 - 12:30

From January 5 to April 13

Course website:

`http://www.cs.mcgill.ca/~cs202`

What this course is about

- This course is an introduction to *computer programming*
- Computer programming: solving problems involving information by means of a computer

What this course is *not* about

- This course is ***not*** about...
  - ...how to use a computer
  - ...how to use software applications
  - ...how to use the Operating System
  - ...how to send e-mail
  - ...how to surf the Web
  - ...how to create Web pages
  - ...how to fix your printer
  - ...how to become a hacker
  - ...how to manage a computer system (installing software, fixing problems, etc.)
- There is no course in Computer Science about how to use computers, in the same way that there is no course in Mechanical Engineering that teaches how to drive a car or operate some machinery.

Objectives

- To learn:
    - ...a methodology to understand and solve problems involving information
    - ...how to think computationally
    - ...how to create simple algorithms
    - ...how to design and implement computer programs using the Java programming language
    - ...how to solve problems in an Object-Oriented manner
- This is neither a "computers course" nor a "Java course."

Fundamental concepts

- Algorithms: An algorithm is a well-defined procedure to solve a problem

- Programming Language: A formal language used to express algorithms

- Programs: The realization of some algorithm in a programming language

Why is computer programming useful

- General benefits
  - Introduces a structured way of thinking, analysing and solving problems

- Applications
  - Engineering and Physical sciences: modelling and simulation
  - Biological sciences: Bioinformatics, Eco-system modelling
  - Geography, Enviromental Studies and Urbanism: Geographic Information Systems
  - Economics: Economic forecasting and analysis, Economic modelling
  - Management: Databases, Information Systems, Process optimization
  - Software development

Who is this course for

- Required for:
  - Major in Software Engineering
  - Major in Computer Engineering
  - Major in Electrical Engineering
  - Minor in Computer Science
  - ...others
- Anyone interested in learning how to develop software

Prerequisites

- An upper-level CEGEP Mathematics course or equivalent
- Logical thinking: being able to reason, to deduct and to infer
- Familiarity with using computers:
  - Editing and saving text files
  - File system: using directories/folders (navigating, copying files, etc.)

Is this course easy?

- No
- This course is considered easy by approximately 5% to 10% of previous students
- The workload is heavy, specially after assignment 2.
- The exams are long
- Course withdrawal: please consult the Undergraduate Course Calendar

Grading system

- The marks will be divided as follows:
  - Assignments: 25%
  - Midterm: 20%
  - Final: 55%
- Assignments:
  - INDIVIDUAL
  - There are 6 assignments
  - Only the best 5 assignments count
  - To be submitted electronically through WebCT
- Midterm: covers all topics up to the day before the exam
- Final: covers all topics

Plagiarism

- All coursework must be done INDIVIDUALLY

- You may not work in groups: if you need help, contact a TA or instructor

- Each assignment and exam must be marked with your full name and student id

- By putting your name and id you are stating that the assignment is entirely your own work

- Students who put their name on programs, modules, or parts of programs that are not entirely their own work will receive a mark of 0 for that assignment, and this mark will be counted as one of the 5 assignments marks included in the final grade. In addition, the students involved may be referred to the appropriate Associate Dean who will assess the need for further disciplinary action.

Office hours

- Where: McConnell Engineering Building, room 202
- When: Wednesdays from 2:00pm to 4:00pm
- ...or by appointment (e-mail)
- ...but you can come by (almost) anytime
- E-mail: eposse@cs.mcgill.ca
- Teaching Assistants (TAs): office hours TBA
- Treat the TAs respectfully

What you will need

- The textbook: Java Software Solutions by John Lewis and William Loftus
- Available at the McGill Bookstore (you may use old and used editions.)
- Access to a computer:
  - Either at home
  - ...or at the Trottier labs (Trottier Building, 3rd floor)
  - ...or anywhere else
- Software:
  - The Java Software Development Kit (j2sdk)
  - An IDE (Integrated Development Environment)

If you use the Trottier labs

- Located at the third floor of the Lorne M. Trottier Building

- All machines are Linux or Unix boxes (no Windows or Macintosh computers)

- Openning an account: (only if you are officially registered)
  - Enter username and password
    - username: newuser
    - password: newuser
  - Answer what you are asked
  - If you need extra help, ask for the consultant

- These machines have already installed the j2sdk and NetBeans, and IDE

- To learn about Linux/Unix, there will be seminars next week at the beginner and intermmediate levels.

If you use another machine

- You need to install the j2sdk:
  - It comes with the book
  - It can also be downloaded for free from
    `http://java.sun.com`
    - Download J2SE, Desktop, any version after 1.3.1
- You need to install an IDE
  - For example a free IDE for Windows is JCreator LE, which can be downloaded from:
    `http://www.jcreator.com`
- Install the IDE after installing the j2sdk

Hints for not suffering in this course

- READ CAREFULLY
- Don't wait until the last minute to do your assignments
- Do not copy any part of anyone else's assignment (current or past students)
- Do not work in groups. If you have difficulties, contact the instructor or the TAs.
- Do not expect to be given every single detail. Expect to deduce things on your own.
- Experiment!

Computers and Information

- What is a computer?
- How computers work?
- How is information stored/represented in a computer?

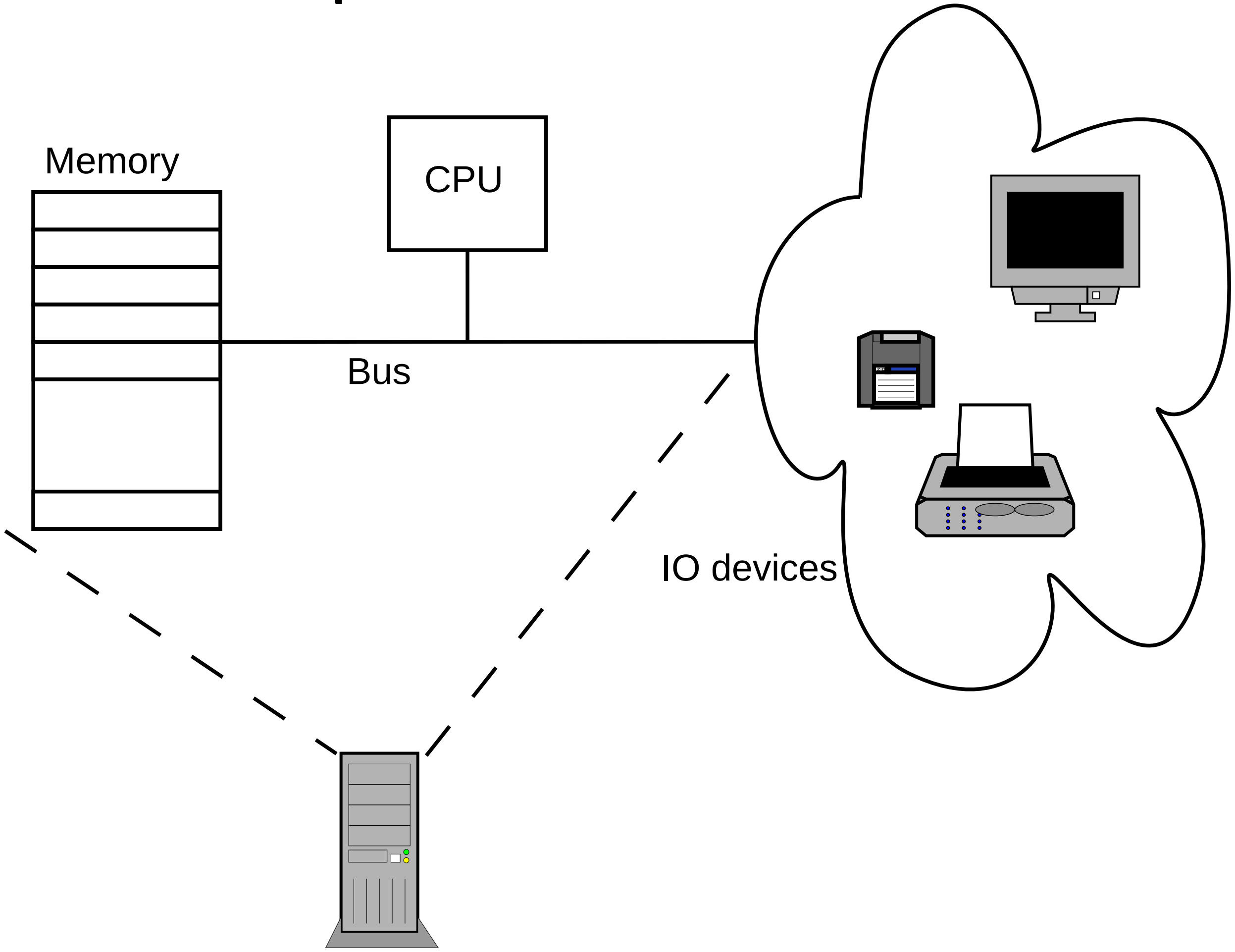
Computers and Information

- A computer is a machine that can perform many different tasks
- ...but the tasks are not predefined
- A computer is a machine which can execute instructions which we give to it
- Therefore, if we can change the set of instructions we can tell the computer to do different things

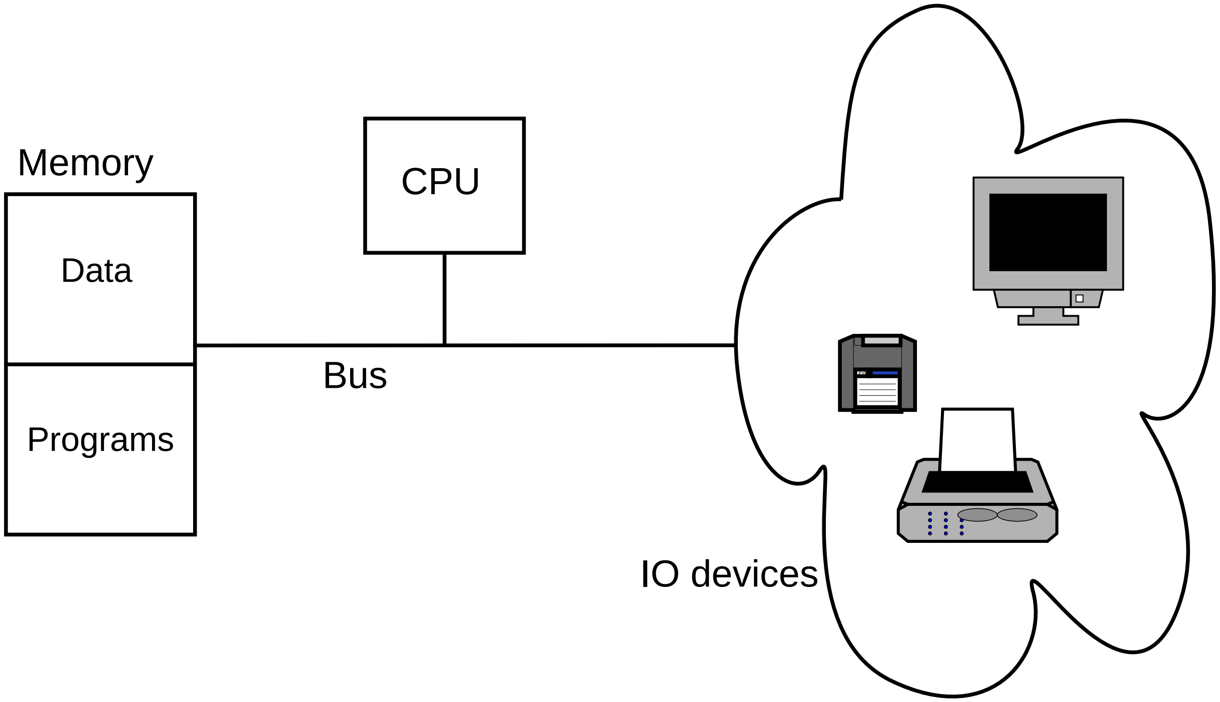
Computers and Information

- Hardware vs Software
  - Hardware: circuits
  - Software: programs
    - Application programs
    - Operating System
- Computer components (Hardware):
  - CPU (Processor)
  - Memory (RAM/ROM/etc.)
  - Input-Output Devices (IO, Keyboard, Screen, Mouse, Printer, etc.)
  - Note: Disks (Hard Disks, CDs, etc.) are IO devices which store data, so they can be seen also as a kind of memory
  - Bus

Computers and Information



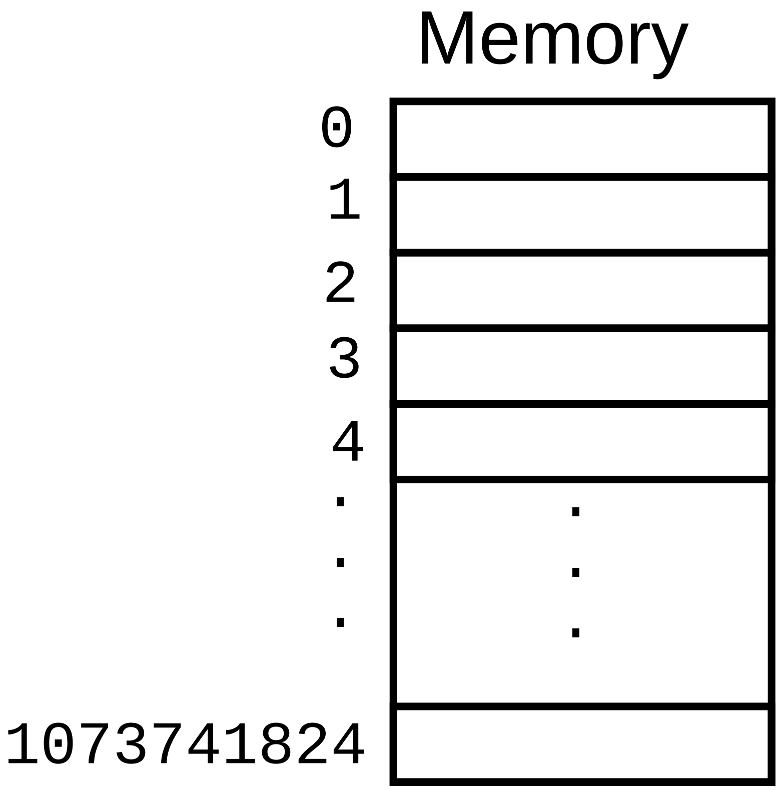
Computers and Information



Memory, data and programs

- Memory:
  - Memory is a very long (but finite) list of *cells* or *memory locations*
  - Each cell is assigned a unique *address* (a natural number)
  - Each cell contains some piece of information (of fixed size)
  - Some cells contain just data
  - Other cells contain instructions for the processor
- Programs
  - A program is a sequence of instructions
  - A program can be stored in memory
  - Programs manipulate the data which is stored in other memory locations
  - Programs are data which is *executable* by the processor (Von Neumann Architecture)

Memory, data and programs



Program execution

- The CPU keeps track of the program which it is executing
- The CPU takes each program instruction (one at a time) from memory, ...
- ... and executes the instruction...
- ...which may involve:
  - making an arithmetic computation
  - reading from or writing to memory
  - reading from or writing to an IO device
  - other operations (changing the next instruction to be executed)
- The traffic of data between the components is through the bus

Data representation

- Data is stored in memory
- Memory cells store numbers
- Numbers represent different types of information:
  - letters
  - text
  - graphics/pictures/images
  - sound
  - movies
  - structured data (e.g. databases, tables, etc.)
  - mathematical functions
  - programs

Data representation

- How are numbers represented in memory?
  - A computer is an electronic device made out of wires
  - Wires have a voltage
  - We can think of the voltage of a wire as the *state* of the wire
  - Different voltages can represent different values

- To simplify things, digital circuits have wires with only two possible voltages (e.g. 0 and +3V).

- Hence a single digital wire can represent something that has two possible values: a *bit* (true/false, on/off, up/down, yes/no, ...)

- The bit is the fundamental unit of information: 0 and 1

Data representation

- To represent more complex things, we can form sequences of bits: 000101, 1101001, 00, 111111111111, 1010101010, ...

- Bit sequences represent binary numbers: numbers in base 2:

  - 0 is 0
  - 1 is 1
  - 2 is 10
  - 3 is 11
  - 4 is 100
  - 5 is 101
  - ...

- Binary numbers are ordinary numbers which are written with only two digits (0 and 1) instead of ten (0 to 9).

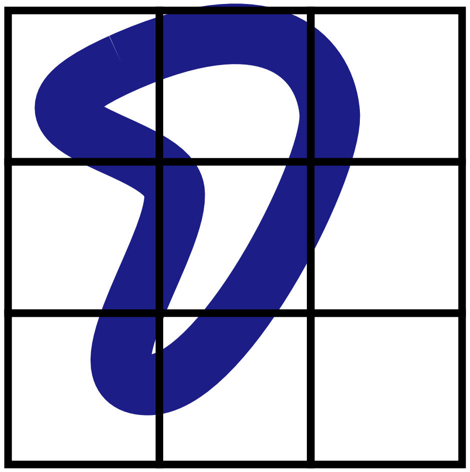
Data representation

- Bit sequences can represent other things: e.g. letters
  - 'a' is 10001001
  - 'b' is 10001010
  - 'c' is 10001011
  - ...
- And therefore text: "bca" is 100010101000101110001001

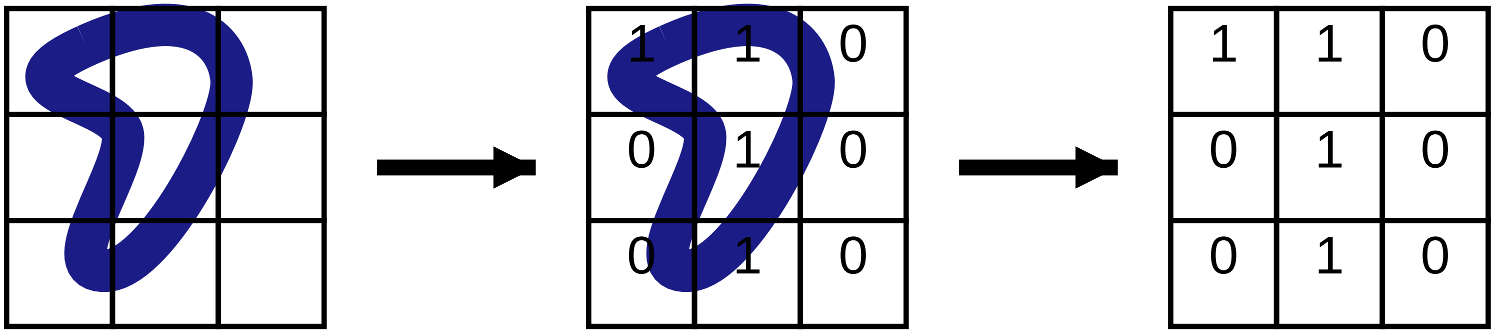
Data representation

- They can also represent images

Data representation



Data representation



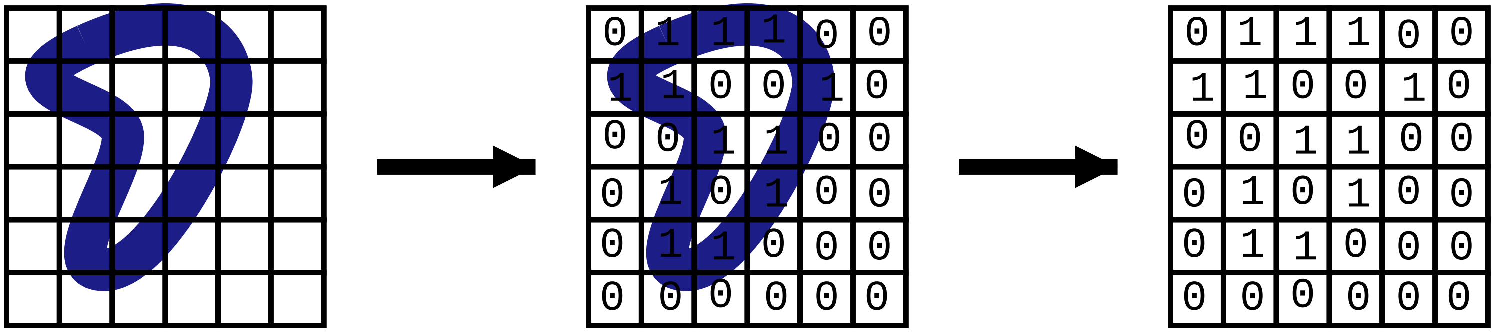
1	1	0
0	1	0
0	1	0

110010010

or

100111000

Data representation



0	1	1	1	0	0
1	1	0	0	1	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	0	0	0
0	0	0	0	0	0

011100110010001100010100011000000000

or

010000110110101010101100010000000000