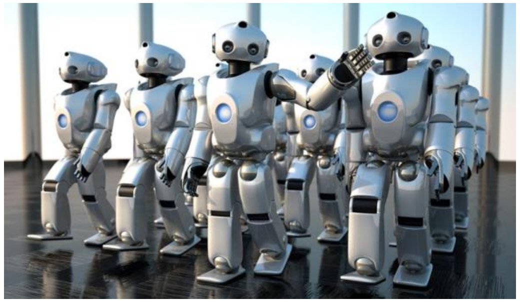
Algorithms and Data Structures

MTE 140 – Winter 2018

Dr. Robert Amelard

ramelard@uwaterloo.ca

On Algorithms and Data Structures



- By the time you've sorted out a complicated idea into little steps that even a stupid machine can deal with, you've certainly learned something about it yourself. – Douglas Adams

- This is where the world is going: direct access from anywhere to any type of data, whether it's a small piece of data or a small answer but a long algorithm to create that answer. The user doesn't care about this. – Hasso Plattner

- Algorithms don't do a good job of detecting their own flaws. – Clay Shirky

[GoodReads. Online, 2016. http://www.goodreads.com/; BrainyQuote. Online, 2016. http://www.brainyquote.com]

http://randomwallpapers.net/nothing-in-the-way-of-creativity_w566019

Who Am I?

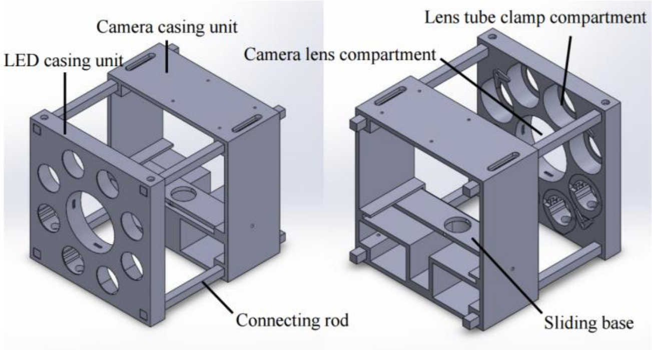
Dr. Robert Amelard (Rob)

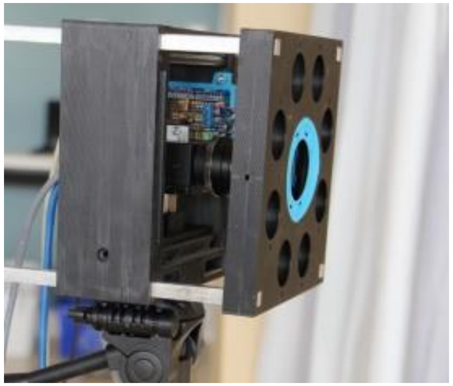
- PhD, Systems Design Engineering (UW)
- Bachelor of Software Engineering (UW)
- Worked at large and small software firms

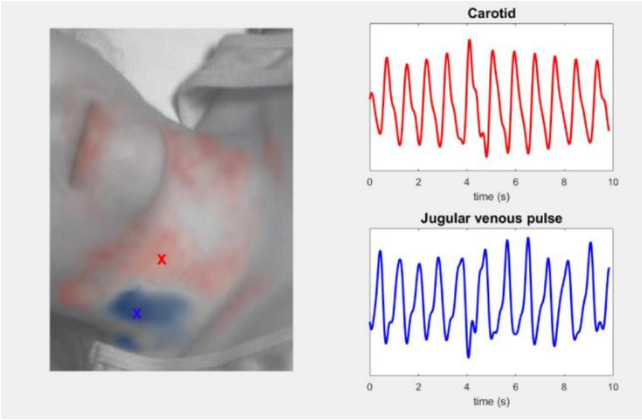
Google uncharted TD

CHRISTIE AGFA Agfa HealthCare

- Research in biomedical imaging, machine learning and image processing







My Core Principles

- Strong Link with Practice
  - Set you up for interviews & co-op jobs

eclipse

slack

Guest lecture

- Learning from Failures
  - Elon Musk, James Dyson, many other greats…

3 failed launches

5,126 prototypes

Lab “conversations”

“You must remove any sort of criticism. A revolutionary suggestion often sounds stupid and you don’t want people who make cynical remarks. It’s a matter of having the right attitude -- humble, curious, determined, willing to fail and try. (I hire people who) embrace the fact that failure is interesting.” – James Dyson

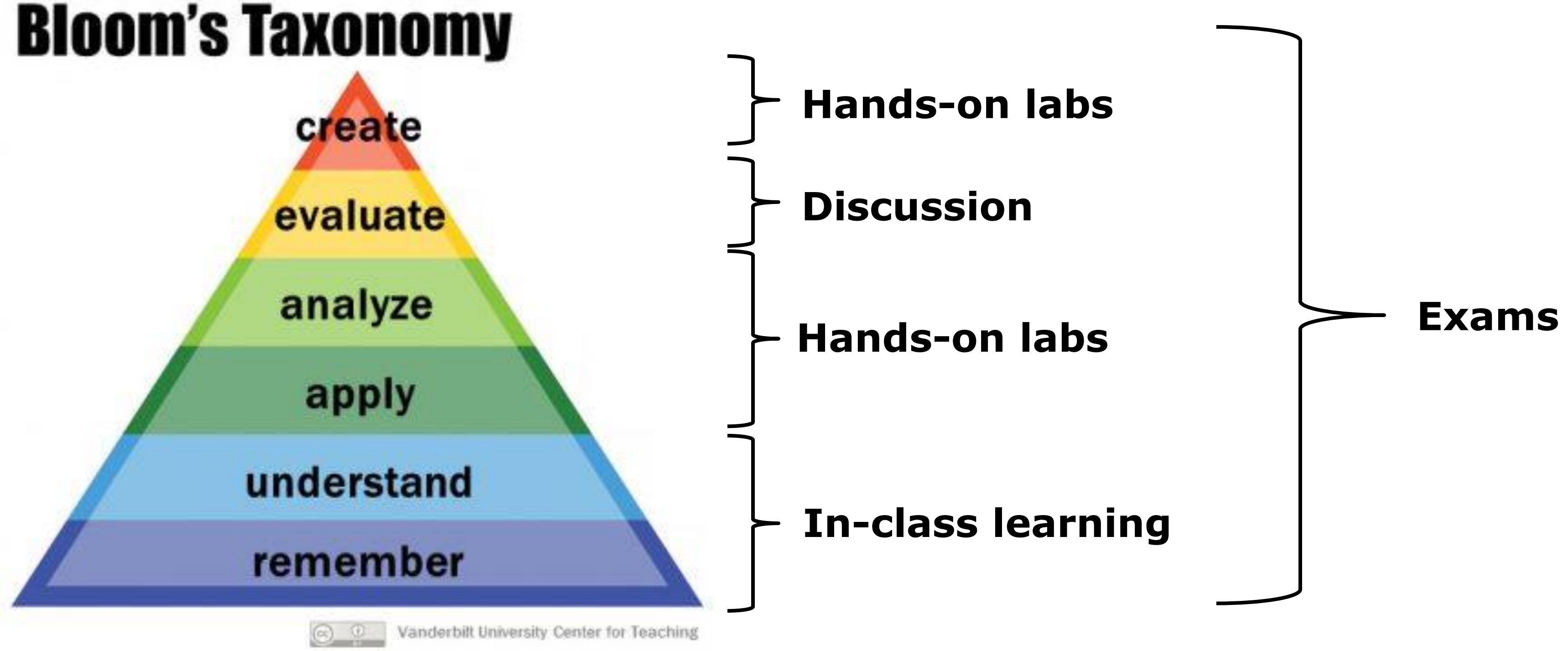
- Approachable and Available

Office hours

Email policy

Feedback

MTE140 Course Model



Hands-on labs

Discussion

Hands-on labs

In-class learning

Exams

Next: Course Syllabus

Syllabus /1

- **How to Contact Me?**
  - **Dr. Robert Amelard (Rob)**
  - **Email**: ramelard@uwaterloo.ca
  - **Slack**: @Rob (instructor) via Slack
  - **Office Hours**: Fri 1:30-2:30pm, EC4-2013

- **How to Contact TAs?**

  - **Audrey Chung** agchung@uwaterloo.ca
    Slack:@Audrey (TA)
  - **Mohsen Ghanbari** mohsen.ghanbari@uwaterloo.ca
    Slack:@Mohsen (TA)
  - **Mahzar Eisapour** eisapour@uwaterloo.ca
    Slack:@Mahzar (TA)
  - **Lin Nei** l7nie@uwaterloo.ca
    Slack:@Lin (TA)

Syllabus /2

- **Course Description:**
  - Algorithms and Data Structures emphasizes the following topics: structured software design data structures, abstract data types, recursive algorithms, algorithm analysis and design, sorting and searching, hashing, and problem-solving strategies.

- **Selected Follow-Up Courses:**
  - MTE 241 Introduction to Computer Structures & Real-Time Systems
  - SYDE 322 Software Design
  - ECE 325 Microprocessor Systems and Interfacing for Mechatronics Engineering
  - And many more…

Syllabus /3

- **Learning Objectives:**
  - Students will learn to analyze, implement and use various data structures and data-management techniques in a variety of applications. This includes:
    - Determining a suitable data structure to solve a specific algorithmic problem;
    - Performing rigorous complexity analyses of simple algorithms and data structures;
    - Comparing different data-structuring techniques from the point of view of time and memory requirements; and
    - Implementing data structures correctly and efficiently in one or more high-level programming languages.

Syllabus /4

- **Course Information on LEARN:**
  - https://learn.uwaterloo.ca WATERLOO LEARN
  - We do not authorize the posting of the course materials outside of LEARN to any other web site.
  - **Regularly check the course web site on LEARN for announcements and deadlines.**

- **Course Discussions:**
  - **Slack:** https://www.slack.com slack

    (Bonus 5%)

Syllabus /5

- **Learning Methodology:**

  a) Attend the interactive lecture, take part in activities during class time, and write detailed class notes; write down any uncertainties that you may have

  b) During study time, complete the assigned activities; also, complete practice exercises and supplementary readings from the course textbook

Syllabus /6

- **Learning Methodology Continued:**

  c) Attend the lab, reflect on the material from class, and complete what is assigned

  d) Using online discussion forums or during office hours, ask for clarifications

  e) Before exams, review your assignment submissions and all your notes

Syllabus /7

- **Course Textbook:**
  - A. Wong et al: Data Structures and Algorithms in a Nutshell. University of Waterloo, ISBN: 281000002061B

$20

- **Additional Reading:**
  - M. Main and W. Savitch: Data Structures and Other Objects Using C++. 4th Edition, Prentice Hall, ISBN: 0132129485
  - W. Savitch and K. Mock: Absolute C++. 5th Edition, Addison Wesley, ISBN: 013283071X

Optional; for your benefit only

Syllabus /8

- **Grading Scheme:**

Lab Assignments	**25%**
Midterm Exam	**25%** **(Feb 15 8:30-10:20AM)**
Final Exam	**50%**
Slack Participation	**5% (bonus)**

- **Exam Policy:**
  - A missed midterm exam will receive a mark of 0, unless there is a valid documented reason. If a documented reason is provided for missing the midterm, its weight will be applied to the final exam.
  - If a student is unable to write the final exam, they need to notify the instructor at least 24 hours before the exam. No INC grade will be awarded without proper notification and documentation.

Syllabus /9

- **Being On Time For Lectures and Quizzes:**
  - Being on time is a small but very important mental discipline that you need to develop to ensure that you are not late for things beyond your University life (e.g., flights, interviews, project presentations, meetings with clients or investors).

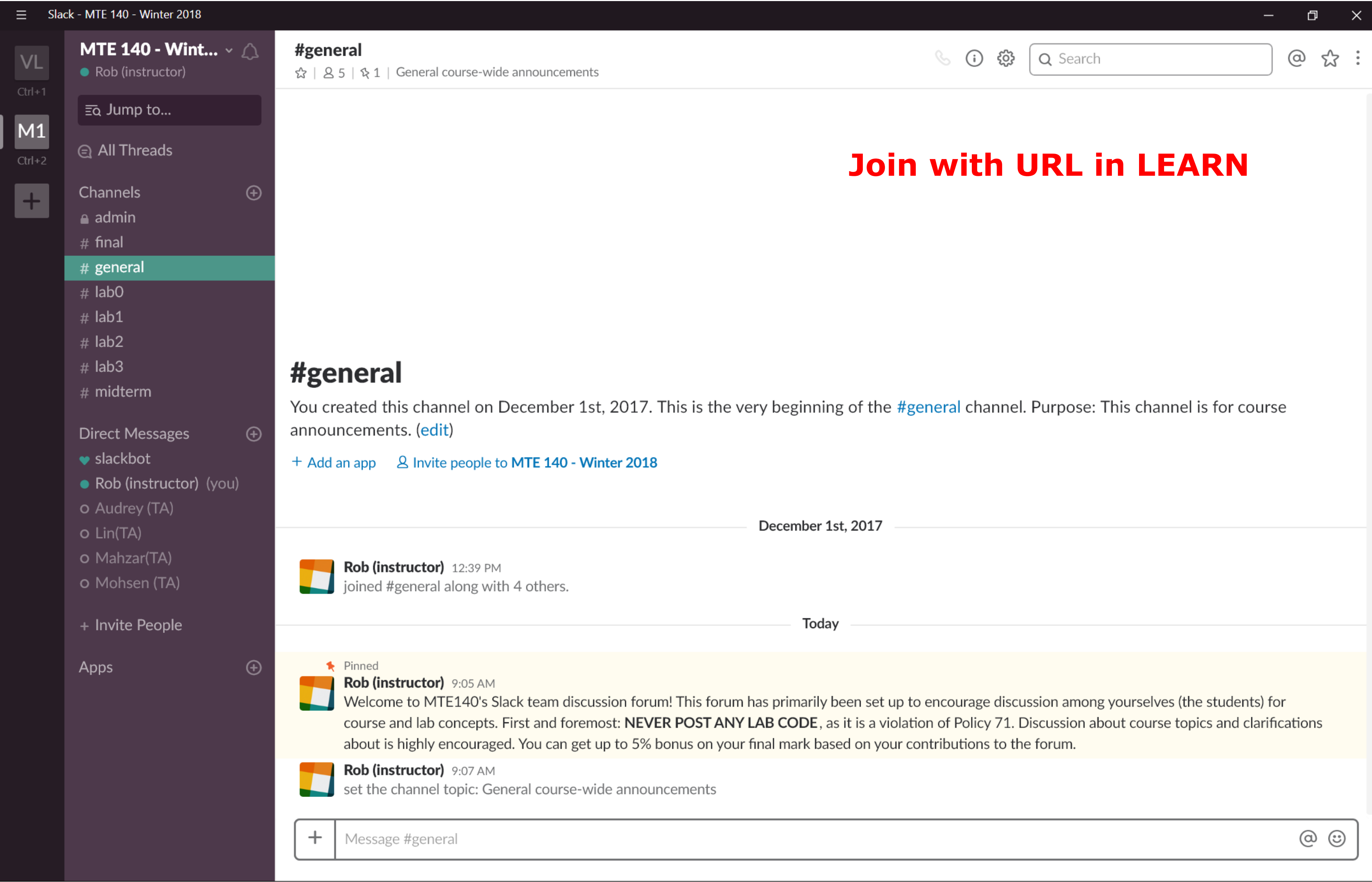
Syllabus /10

- **Labs:**
  - The labs will include practical algorithms and data structures problems related to the topics covered in lectures.
  - These will help hone your higher order thinking (Bloom’s taxonomy).
- **Lab Assignments:**
  - Lab 0: individual/discussion-based; get familiar with C++
  - Lab 1-3: pre-assigned pairs; covers a specific topic from lecture
    - Lab 1: Lists
    - Lab 2: Stacks/Queues
    - Lab 3: Trees
  - Late labs: 20%/day

Syllabus /11

- Slack Discussion
  - The purpose of this message forum is twofold
    - □ (1) to expose you to tools found in industry, and
    - □ (2) to encourage collaborative discussion between peers.
  - This forum will be moderated by the course instructor and TAs, however active discussion by them will be minimal.
  - If you want help from a course TA, either tag them on your message (using the @symbol), or direct message them.
  - You may gain up to 5% bonus marks for your contributions in Slack. That is, if you frequently help other students with their questions, you can gain an extra 5% to your final grade

Syllabus /12



Join with URL in LEARN

#general

You created this channel on December 1st, 2017. This is the very beginning of the #general channel. Purpose: This channel is for course announcements. (edit)

December 1st, 2017

Rob (instructor) 12:39 PM
joined #general along with 4 others.

Today

Pinned
Rob (instructor) 9:05 AM
Welcome to MTE140's Slack team discussion forum! This forum has primarily been set up to encourage discussion among yourselves (the students) for course and lab concepts. First and foremost: **NEVER POST ANY LAB CODE**, as it is a violation of Policy 71. Discussion about course topics and clarifications about is highly encouraged. You can get up to 5% bonus on your final mark based on your contributions to the forum.

Rob (instructor) 9:07 AM
set the channel topic: General course-wide announcements

Syllabus /13

- **Graded Material Delivery:**
  - Labs will be returned by email, exams will be returned in class.

- **Remarking Requests:**
  - “Lab Conversations” **(Learning from Failures)**
    - If you get a mark below 15/20 on a lab, you may have a “lab conversation” with the marking TA. If you **demonstrate a clear understanding of the lab concepts and why your program failed**, you may regain up to half of the lost marks.
  - Exam Remarking Requests
    - When the exams are returned, you may attach a note to your exam clearly stating the questions that you want remarked. Include any supporting evidence for your case.

Syllabus /14

- **Course Schedule:**
  (subject to change)
  - Week 1: Introduction
  - Week 2: Linked Data Structures
  - Week 3: Lists
  - Week 4: Recursion
  - Week 5: Algorithmic Analysis
  - Week 6: Stacks and Queues
  - Week 7: Midterm Week
  - Week 8: Trees
  - Week 9: More on Trees
  - Week 10: Graphs
  - Week 11: More on Graphs, Sorting
  - Week 12/13: More on Sorting, Hashing

Syllabus /15

- **In-Class Discipline (for you):**
  - Students are encouraged to attend all lectures, but are required not to be disruptive during lectures out of respect for their classmates and for the instructor.
  - Disruptive behaviour includes loudly talking with people next to them, texting on a mobile phone, playing YouTube videos, Facebook and Twitter updates, etc.

Syllabus /16

- **Academic Integrity:**
  - In order to maintain a culture of academic integrity, members of the University of Waterloo community are expected to promote honesty, trust, fairness, respect and responsibility. Check the Office of Academic Integrity's website for more information,
  - http://uwaterloo.ca/academic-integrity/

- **Grievance:**
  - A student who believes that a decision affecting some aspect of his/her university life has been unfair or unreasonable may have grounds for initiating a grievance. Read Policy 70, Student Petitions and Grievances, Section 4,
  - http://www.adm.uwaterloo.ca/infosec/Policies/policy70.htm
  - When in doubt please be certain to contact the department’s administrative assistant who will provide further assistance.

Syllabus /17

- **Academic Discipline:**
  - A student is expected to know what constitutes academic integrity to avoid committing academic offenses and to take responsibility for his/her actions.
  - A student who is unsure whether an action constitutes an offense, or who needs help in learning how to avoid offenses (e.g., plagiarism, cheating) or about "rules" for group work/collaboration should seek guidance from the course professor, academic advisor, or the undergraduate associate dean. For information on categories of offenses and types of penalties, students should refer to Policy 71, Student Discipline,
  - http://www.adm.uwaterloo.ca/infosec/Policies/policy71.htm

  - For typical penalties check Guidelines for the Assessment of Penalties,
  - http://www.adm.uwaterloo.ca/infosec/guidelines/penaltyguidelines.htm

Syllabus /18

- **Slack Usage:**
  - Please note that posting partial or complete assignment solutions to Slack in a public post before the assignment deadline is a violation of Policy 71 - Student Discipline.
  - As stated in this policy: "Students are responsible for demonstrating behaviour that is honest and ethical in their academic work. Such behaviour includes: [...] Preventing their work from being used by others, e.g. not lending assignments to others, protecting access to computer files.“
  - Each instance of violation of Policy 71 will have to be treated as academic misconduct, and corresponding penalties as prescribed by the Faculty of Engineering will have to be applied.
  - Also note that Slack are not complaint forums (e.g., complaints about assignment marking). If you have a concern about anything to do with the course, please address your concern directly with the course TA or with the instructor, as appropriate.

Syllabus /19

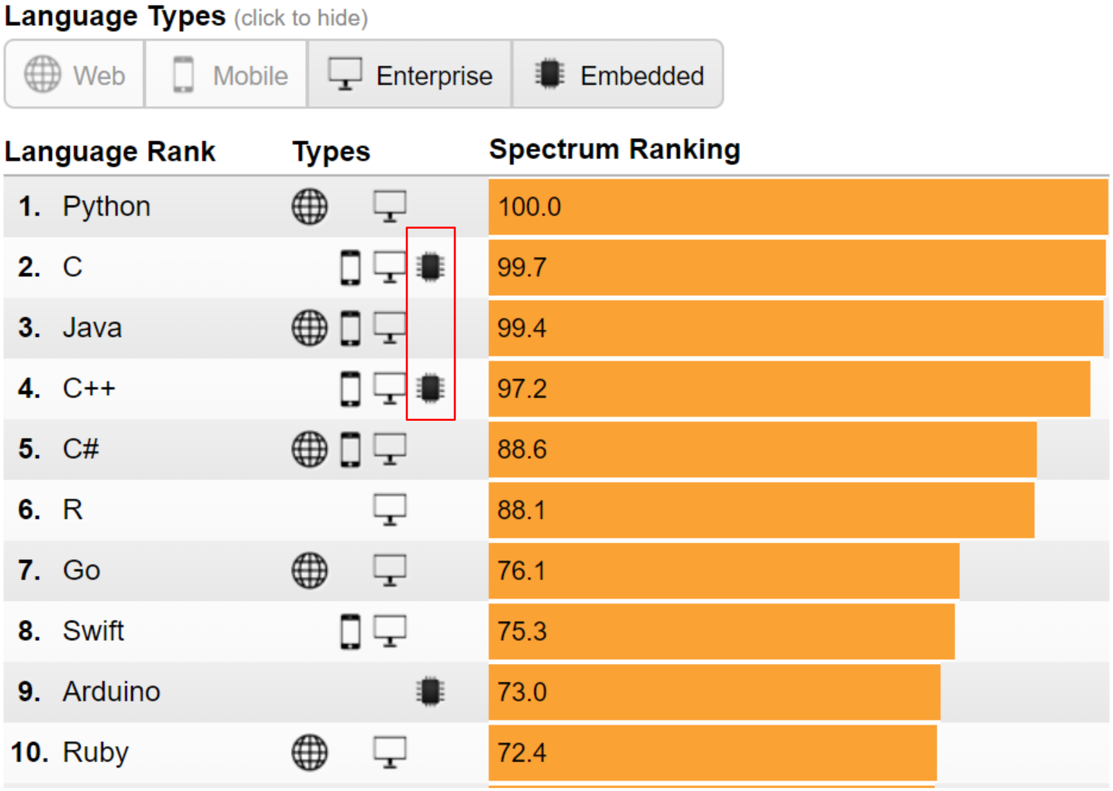
- **Appeals:**
  - A decision made or penalty imposed under Policy 70, Student Petitions and Grievances (other than a petition) or Policy 71, Student Discipline may be appealed if there is a ground. A student who believes he/she has a ground for an appeal should refer to Policy 72, Student Appeals,
  - http://www.adm.uwaterloo.ca/infosec/Policies/policy72.htm
- **AccessAbility Services:**
  - AccessAbility Services, located in Needles Hall, Room 1401, collaborates with all academic departments to arrange appropriate accommodations for students with disabilities without compromising the academic integrity of the curriculum.
  - If you require academic accommodations to lessen the impact of your disability, please register with them at the beginning of each academic term.

Syllabus /20

- Tutoring available to ALL students
  - Office hours
  - Tutoring in residences
  - Evening sessions in WEEF Lab (E2 1310) from Monday through Thursday at 6:30 – 8:30 pm
  - WEEF TAs

Course Tools

- Rooted in relevance
  - Labs in **C++**
  - **Eclipse IDE**
  - **Slack**



Language Rank	Types	Spectrum Ranking
1. Python	🌐 🖥	100.0
2. C	📱 🖥 ▣	99.7
3. Java	🌐 📱 🖥	99.4
4. C++	📱 🖥 ▣	97.2
5. C#	🌐 📱 🖥	88.6
6. R	🖥	88.1
7. Go	🌐 🖥	76.1
8. Swift	📱 🖥	75.3
9. Arduino	▣	73.0
10. Ruby	🌐 🖥	72.4

Note on Course Content

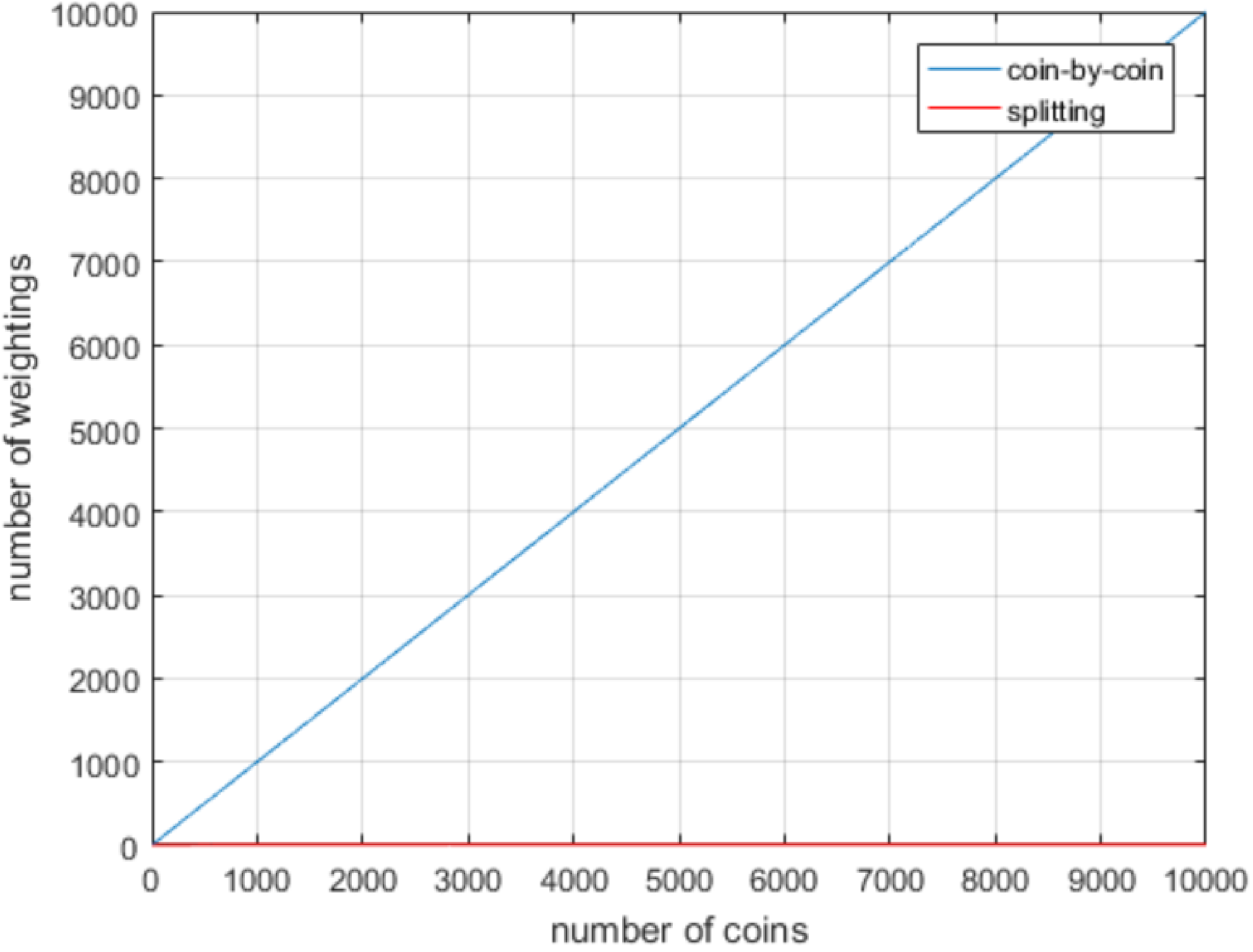
- Foundational in many aspects of software
  - Highly searchable online
  - Use to your advantage for troubleshooting etc.

A Decision-Making Problem

- **Coin-Weighing Problem:**
  - Assume that we are given nine coins
  - All coins look the same, but one of them is lighter and a counterfeit
  - Using a balance scale – see on the right – how many weightings are required to find the counterfeit coin?
  - What if you are given 10,000 coins to find

A Decision-Making Problem

- **Co**
  Pr



Understanding Algorithm Performance

- **Algorithm:**
  - Determines the number of operations executed

Algorithm/data structure design
****MTE140**

- **Coding language, compiler, and architecture:**
  - Determine the number of machine instructions executed per operation

High-level vs Low-level

- **Processor and memory system:**
  - Determine how fast the instructions are executed

Embedded vs Desktop

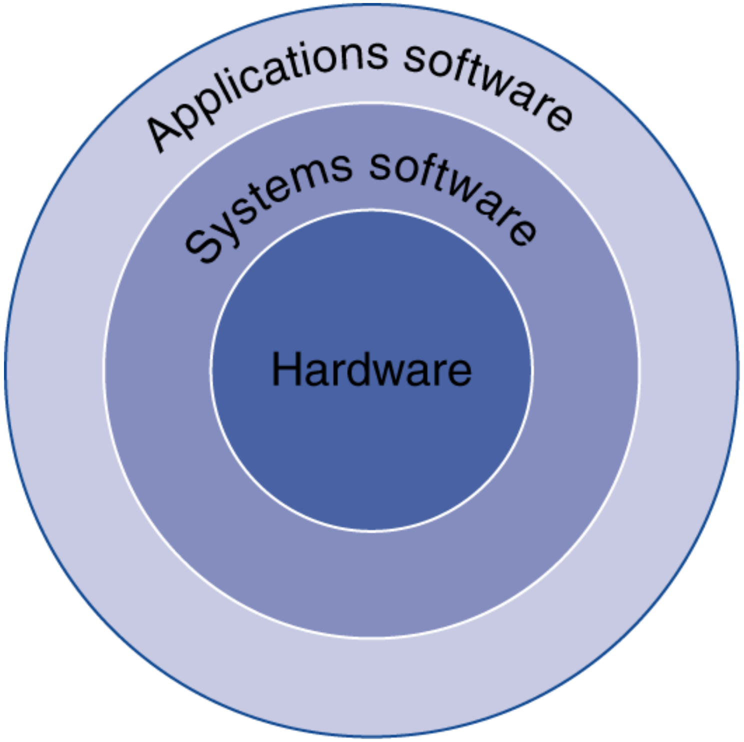
- **I/O system (including OS):**
  - Determines how fast the I/O operations are executed

System design

- **Instruction set architecture (ISA)**
  - The specification of the hardware/software interface

System design

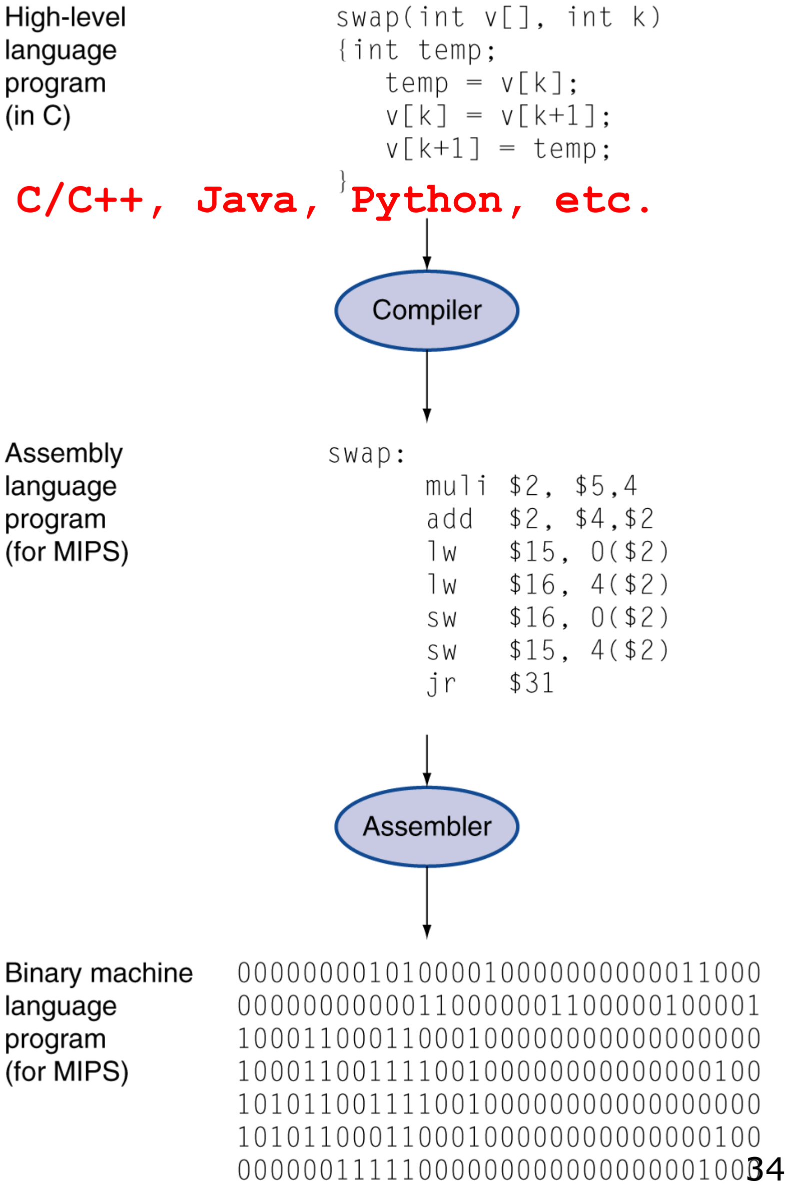
Behind the Curtain



- **Application software**
  - Written in high-level language (HLL)
- **System software**
  - Compiler: Translates HLL code to machine code
  - Operating System: service code
    - □ Handling input/output
    - □ Managing memory and storage
    - □ Scheduling tasks and sharing resources
- **Hardware**
  - Processor, memory, and I/O controllers

Levels of Program Code

- **High-level language**
  - Level of abstraction closer to problem domain
  - Provides for productivity and portability
- **Assembly language**
  - Textual representation of instructions
- **Hardware language**
  - Binary digits (bits)
  - Encoded instructions and data



```
swap(int v[], int k)
{int temp;
   temp = v[k];
   v[k] = v[k+1];
   v[k+1] = temp;
}
```

C/C++, Java, Python, etc.

```
swap:
      muli $2, $5,4
      add  $2, $4,$2
      lw   $15, 0($2)
      lw   $16, 4($2)
      sw   $16, 0($2)
      sw   $15, 4($2)
      jr   $31
```

```
00000000101000010000000000011000
00000000000110000001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
```

Acknowledgements

- Dr. Igor Ivkovic (slides)

Next Steps

- **Read:**
  - Chapter 1 (Introduction) from the course handbook