UBC's Vancouver campus is located on the traditional, ancestral, unceded territory of the Musqueam People.
Today's learning goals

- By the end of this module, you should be able to:
  - Give an example of how we can apply formal reasoning to a simple, real-world task.
  - Give an example of how a computational solution to this simple task might go wrong.
  - Describe the four “big questions” that we will start answering in CPSC 121.
Our shared work to resist anti-Black violence follows from the intellectual, emotional, and creative labours of Black intellectuals, activists, scientists, artists, designers, writers, poets, curators, illustrators, filmmakers, and cultural producers. They form a critical part of our collective learning environment as students, faculty, and professional staff.

**Dorothy Vaughan**
Taught herself and other women from NACA (called human computer) programming languages to prepare them for the transition to machine computers.

**Jerry Lawson**
He led the development of a console, that used swappable game cartridges, replacing ROM storage that was soldered onto the game hardware.
Let's try sorting

- Activity: an algorithm to order students by day of birth.

How many swaps did you need to make if you did not make any mistake?

https://www.cs.ubc.ca/~mochetti/dates/dates.html
Computing $n(n-1)/2$ using DrRacket:

```
(define (how-many-swaps n)
  (/ (* n (- n 1)) 2))
```

Welcome to DrRacket, version 6.2.1 [3m].
Language: Advanced Student; memory limit: 256 MB.
> (how-many-swaps 5)
10
> (how-many-swaps 1000)
499500
> (how-many-swaps 1000000)
499999500000
> |
Computing $n(n-1)/2$ using Java:

```java
import java.io.*;

public class Compute {
    public static void main(String[] args) {
        int n = Integer.parseInt(args[0]);
        System.out.println(n * (n-1) / 2);
    }
}
```

<table>
<thead>
<tr>
<th>Command</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>poirot&gt; java Compute 5</code></td>
<td>10</td>
</tr>
<tr>
<td><code>poirot&gt; java Compute 1000</code></td>
<td>499500</td>
</tr>
<tr>
<td><code>poirot&gt; java Compute 1000000</code></td>
<td>-364189984</td>
</tr>
</tbody>
</table>
Questions to ponder:

- Questions about this exercise:
  - How can we prove that \( n(n-1)/2 \) is the largest number of swaps needed for \( n \) birthdays?
    - Direct Proof or Mathematical Induction
  - Why did our Java implementation print a negative value, but not the Racket implementation?
    - Number representation
1. How can we convince ourselves that an algorithm does what it's supposed to do?

2. How do we determine whether or not one algorithm is better than another one?

3. How does the computer (e.g. Dr. Racket) decide if the characters of your program represent a name, a number, or something else? How does it figure out if you have mismatched " " or ( )?

4. How can we build a computer that is able to execute a user-defined program?
A working computer you will learn about in the labs:
Course activities & infrastructure

- **Course web site**
  
  https://www.students.cs.ubc.ca/~cs-121

- **Canvas**
  
  Used for lab grades and pre-class quizzes.

- **Overleaf** ([https://www.overleaf.com/]())
  
  Used to write assignments and turn them into PDFs.
  
  You can scan handwritten solutions too, but we don't recommend it.

  You also get introduced to LaTeX, which is a very useful (and widely-used) tool for scientific writing.
Course activities & infrastructure

- **Piazza:**
  - For discussions.
  - Please register through Canvas.
  - For daily access, there is a link on the course website.
Course activities & infrastructure

- Gradescope (https://www.gradescope.ca/)
  - Used to
    - submit assignments
    - receive feedback and grades for assignments & midterms.
    - submit regrade requests for assignments & midterms.
  - Gradescope accounts
    - Will be created over the week-end.
    - You will get an email from Gradescope prompting you to set your password.
Course activities & infrastructure

- Clickers
  - [https://www.iclicker.com/](https://www.iclicker.com/)
  - We’re still figuring things out as IT services has been making changes to Canvas’ clicker support until the last couple of days.
  - We will post detailed instructions on Piazza by the end of the week-end.
  - We will first use clickers on Tuesday September 15th.
Grading scheme

- Your final grade will be computed as follows:
  - Assignments (5): 14%
  - Labs (9): 14%
  - Pre-class quizzes (11): 4%
  - Take-home tests (2): 18%
  - Midterm: 14%
    - Friday October 23rd, 2020, 17:00 to 18:15
  - Final Exam: 32%
  - Clicker questions: 2%
  - Participation: 2%
Grading scheme

- To pass the course you must
  - Obtain a total score of at least 50%.
  - Obtain at least 50% on the weighted average of the labs and assignments.
  - Obtain at least 50% on the weighted average of the two take-home tests, midterm and final exam.
Course activities & infrastructure

- Labs
  - Start on Wednesday September 16\textsuperscript{th}, 2020.
  - You \textbf{must} attend the lab you are registered for.
    - Pre-lab work must be done \textbf{before} you get to the lab.

- Tutorials
  - Start on Monday September 13\textsuperscript{th}, 2020.

- Participation
  - The largest of tutorial attendance & weighted average of all other course components.
Course Mechanics

- **Worksheets:**
  - Will be posted on the course web site ahead of time.
  - We will use breakout rooms while you work on them:
    - Form groups of 4 or 5 students.
    - Include the group name in parentheses in front of your name in zoom.
      - E.g. *(baldpeople++)* Patrice Belleville
    - The breakout room manager will put all of you in the same breakout room.
Course activities & infrastructure

- Academic Conduct.
  - Please read the rules in the syllabus and follow them.
  - Not following these rules will be unpleasant for all of us.
  - We referred several academic misconduct cases to the Science Dean’s office in 2019W2.
  - I really, really don’t want to have to go through this again this term (but I will if necessary).
# Detailed Schedule

<table>
<thead>
<tr>
<th></th>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sept 6&lt;sup&gt;th&lt;/sup&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>First class</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sept 13&lt;sup&gt;th&lt;/sup&gt;</td>
<td></td>
<td></td>
<td></td>
<td>Lab 1</td>
<td>Lab 1</td>
<td>Lab 1</td>
<td></td>
</tr>
<tr>
<td>Sept 20&lt;sup&gt;th&lt;/sup&gt;</td>
<td>Lab 1</td>
<td>Lab 1</td>
<td>Lab 2</td>
<td>Lab 2</td>
<td>Lab 2</td>
<td>Lab 2</td>
<td></td>
</tr>
<tr>
<td>Sept 27&lt;sup&gt;th&lt;/sup&gt;</td>
<td>Lab 2</td>
<td>Lab 2</td>
<td><strong>Test 1</strong></td>
<td>Lab 3</td>
<td>Lab 3</td>
<td>Lab 3</td>
<td></td>
</tr>
<tr>
<td>Oct 4&lt;sup&gt;th&lt;/sup&gt;</td>
<td>Lab 3</td>
<td>Lab 3</td>
<td>Lab 3</td>
<td>Lab 4</td>
<td>Lab 4</td>
<td>Lab 4</td>
<td></td>
</tr>
<tr>
<td>Oct 11&lt;sup&gt;th&lt;/sup&gt;</td>
<td><strong>Holiday</strong></td>
<td>Lab 4</td>
<td>Lab 4 Ass 2 due</td>
<td>Lab 5</td>
<td>Lab 5</td>
<td>Lab 5</td>
<td></td>
</tr>
<tr>
<td>Oct 18&lt;sup&gt;th&lt;/sup&gt;</td>
<td>Lab 4</td>
<td>Lab 5</td>
<td>Lab 5</td>
<td></td>
<td></td>
<td></td>
<td><strong>Midterm</strong></td>
</tr>
</tbody>
</table>
## Detailed Schedule [continued]

<table>
<thead>
<tr>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oct 25th</td>
<td>Lab 5</td>
<td>Lab 6</td>
<td>Lab 6 Ass 3 due</td>
<td>Lab 6</td>
<td>Lab 6</td>
<td>Lab 6</td>
</tr>
<tr>
<td>Nov 1st</td>
<td>Lab 6</td>
<td>Lab 7</td>
<td>Lab 7</td>
<td>Lab 7</td>
<td>Lab 7</td>
<td>Lab 7</td>
</tr>
<tr>
<td>Nov 8th</td>
<td>Lab 7</td>
<td></td>
<td>Holiday Ass 4 due</td>
<td>Test 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nov 15th</td>
<td>Lab 8</td>
<td>Lab 8</td>
<td>Lab 8</td>
<td>Lab 8</td>
<td>Lab 8</td>
<td>Lab 8</td>
</tr>
<tr>
<td>Nov 22th</td>
<td>Lab 9</td>
<td>Lab 9</td>
<td>Lab 9</td>
<td>Lab 9</td>
<td>Lab 9</td>
<td>Lab 9</td>
</tr>
<tr>
<td>Nov 29th</td>
<td>Ass 5 due</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Course Learning Outcomes

- After you complete the course, you will be able to:
  - model important problems so that they are easier to discuss, reason about, solve, and test.
  - learn new modeling formalisms more easily.
  - communicate clearly and unambiguously with other CS experts on complex topics.
  - characterize algorithms (CS problem solutions), by proving their correctness or efficiency.
  - critically read proofs: justifying why each step is correct and judging what the proof means.
  - explain how computers work.
First pre-class Quiz

- First pre-class quiz: due Monday September 14\textsuperscript{th} at 19:00.

- Sections to read for the quiz (read \textbf{one} of these):
  - Epp, 4\textsuperscript{th} edition: 2.1 and 2.4.
  - Epp, 3\textsuperscript{rd} edition: 1.1 and 1.4
  - Rosen, 7\textsuperscript{th} edition: 1.1 up to the top of page 6, and 12.3.
  - Rosen, 6\textsuperscript{th} edition: 1.1 up to the top of page 6, and 11.3.
Coming soon...

- Second pre-class quiz: due Wednesday September 16\(^{th}\) at 19:00.
  - Assigned reading for the quiz:
    - Epp, 4th edition: 2.2
    - Epp, 3rd edition: 1.2
    - Rosen, 6th or 7th edition: 1.1 from page 6 onwards.
- Assignment #1 will be available early next week.
  - due Monday September 28\(^{th}\), 2020 at 19:00.