

---

# CPSC 103 Section 911

— Instructor: Jessica Wong —

[jhmwong@cs.ubc.ca](mailto:jhmwong@cs.ubc.ca)

---

# What is this course about?

Introduce you to software design in a way that will let you write high quality programs specifically targeted at manipulating data in some area interesting to you.

# Course Goals

- 9 modules in total
  - 8 modules are skill based and the 9th is a project
  - First eight modules help you build the skills you need for the project
- Project
  - Take some data from a file, read it in, do something to it, and make a visualization of some sort
  - You can use your own data or we have some choices for you
  - Topic has to be appropriate for a school environment but doesn't have to be academic!

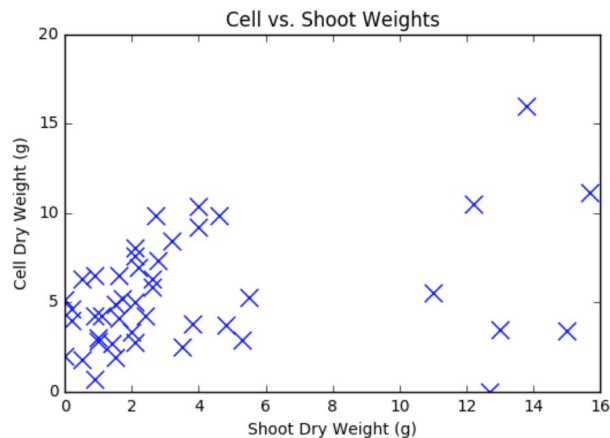
# Project Example

- A former CPSC 103 student was doing research on the relationship between autotrophs and dried eelgrass
- His project focused on graphing his research data
  - You don't need to have research data on hand for this class!

# Project Example

Data file (in CSV form)

Date		quadrat #	epi blade #	epi blade len	epi blade wid	GF/C + tin w	GF/C + tin w	epi_cell_dry	epi foil weig	epi wet weig	wet_epi_we	epiphyte dry	sheath length
20-Jan-16	Jan	1	NA	78.9	0.6	1.0983	1.1004	2.1	9.2	14.2	5	5	
20-Jan-16	Jan	2	NA	72.2	0.6	1.0826	1.0841	1.5	9.2	14.1	4.9	NA	
20-Jan-16	Jan	3	NA	77.9	0.7	1.0708	1.0725	1.7	9.1	14.3	5.2	NA	



# General Structure of Each Module

1. Pre-class assignment
  - a. We use the questions you ask here to drive our class on the next day.
2. Class time
  - a. Discuss commonly asked questions
  - b. Go over concepts
  - c. Work on the worksheet together
  - d. Go through examples
3. Tutorial
4. Worksheet
5. Code Review

# Deadlines

- Mondays and Wednesdays at 10PM are when pre-lecture assignments are due
  - We use these to shape the kinds of things we talk about in class on Tuesday/Thursday
- Sundays at 10PM are when your worksheets and code reviews are due
- Tutorials are generally due 5 days after we finish covering a module
- **See Canvas for a full list of all the deadlines for the course**

# Class Resources

- Canvas
  - Your assignments are listed there
  - Midterms from previous semesters are listed there as practice!
    - Previous midterm 1s listed there now. Midterm 2s will be available later in the semester
- Piazza
  - **Please** ask questions! We like answering questions!
  - If you are unsure about whether your question is appropriate to post for all students, post privately! We (or you) can always set it to public later.
- Syzygy/Jupyter
  - You will code here
  - The files required for code reviews/tutorial or the slides/code files we go through in lecture can all be found there
  - The answer keys for the tutorials will also be found there



# Other Notes

- Test dates have been set! Dates and times listed are based on Vancouver's timezone.
  - Thursday May 28 @ 6PM
  - Tuesday June 9 @ 6PM
- Final exam date TBD by UBC
- Email [cpssc103-admin@cs.ubc.ca](mailto:cpssc103-admin@cs.ubc.ca) for admin questions/concerns
  - E.g., You can't take the test because of class
- If you need to reach me, email me directly at [jhmwong@cs.ubc.ca](mailto:jhmwong@cs.ubc.ca) .  
Canvas mail is often slower and easier to miss.

# Browser Warning

- Our tests and exams will use a combination of Canvas and Jupyter
- Some students have reported issues with seeing images on Canvas when using Safari
- Students who use IE/Edge may run into issues with Jupyter
- We would recommend Chrome or Firefox

# How to Do Well in CPSC 103

- Summer classes are incredibly condensed so we will cover a lot of material very quickly
  - Each class is the equivalent of a week's worth of material
  - Keep on top of your deadlines. Come see us if you feel like you want to talk or get help!
- To get better at programming, you have to do lots of practice
  - The whole course is designed around this concept of practice
  - We'll have lots of time in class to work together on this
- Don't get discouraged if your code doesn't work right away! Debugging is something all computer scientists puzzle over. It's also a great way to think about how code works.
  - Start early. Debugging can sometimes take a very long time

# Tracing Code

```
a = 1
```

```
b = a + 10
```

```
a = a + 10
```

```
a == b
```

```
b = a + b
```

```
a = 100
```

```
a + b
```

## Computer Memory



# Tracing Code

`a = 1` ←

`b = a + 10`

`a = a + 10`

`a == b`

`b = a + b`

`a = 100`

`a + b`

## Computer Memory



# Tracing Code

`a = 1` ←

`b = a + 10`

`a = a + 10`

`a == b`

`b = a + b`

`a = 100`

`a + b`

## Computer Memory

a 1

# Tracing Code

`a = 1`

`b = a + 10` ←

`a = a + 10`

`a == b`

`b = a + b`

`a = 100`

`a + b`

## Computer Memory

a 1

b

# Tracing Code

`a = 1`

`b = a + 10` ←

`a = a + 10`

`a == b`

`b = a + b`

`a = 100`

`a + b`

## Computer Memory

a 1

b 11



# Tracing Code

```
a = 1
```

```
b = a + 10
```

```
a = a + 10 ←
```

```
a == b
```

```
b = a + b
```

```
a = 100
```

```
a + b
```

## Computer Memory

a 1

b 11

# Tracing Code

`a = 1`

`b = a + 10`

`a = a + 10` ←

`a == b`

`b = a + b`

`a = 100`

`a + b`

## Computer Memory

a 11

b 11

# Tracing Code

```
a = 1
```

```
b = a + 10
```

```
a = a + 10
```

```
a == b ←
```

```
b = a + b
```

```
a = 100
```

```
a + b
```

## Computer Memory

a 11

b 11

# Tracing Code

`a = 1`

`b = a + 10`

`a = a + 10`

`a == b`

`b = a + b` ←

`a = 100`

`a + b`

## Computer Memory

a 11

b 11

# Tracing Code

`a = 1`

`b = a + 10`

`a = a + 10`

`a == b`

`b = a + b` ←

`a = 100`

`a + b`

## Computer Memory

a 11

b 22

# Tracing Code

`a = 1`

`b = a + 10`

`a = a + 10`

`a == b`

`b = a + b`

`a = 100`



`a + b`

## Computer Memory

a 11

b 22

# Tracing Code

`a = 1`

`b = a + 10`

`a = a + 10`

`a == b`

`b = a + b`

`a = 100`



`a + b`

## Computer Memory

a 100

b 22

# Tracing Code

```
a = 1
```

```
b = a + 10
```

```
a = a + 10
```

```
a == b
```

```
b = a + b
```

```
a = 100
```

```
a + b ←
```

## Computer Memory

a 100

b 22