# Programming, Problem Solving, and Algorithms

CPSC203, 2022 W2

# Announcements

- **Lab2 is this week**
  - It’s about Lists and Dictionaries!
- **Test 1 results will be released when everyone is done!**
  - Sorry about the snow-related disruptions!
- **Test 2 is next week**
  - It will be on content discussed last week (Class 2A, 2B, POTW 2), and practiced this week (Lab 2):
- **Setup your machines**
  - This week you will setup your machines to run the software stack!

# When to do what in CPSC 203

X – This week's content
O – Previous week's content

| | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday | Sunday |
|---|---|---|---|---|---|---|---|
| Pre-class Videos | X | | X | | | | |
| Classes | | X | | X | | | |
| Learning Log | | | | | X | | |
| Problems of the Week | X | X | X | X | X | | |
| Labs | O | O | | | X | | |
| Practice Test and Test | O | O | O | O | | | |

Note: this is a "suggested" schedule. All due dates are on Friday at 6 PM (with a 48-hour grace period).

3 term projects are not on this schedule as they span multiple weeks.

# Today's Plan...

1. Announcements! (10 mins)
2. Quick re-cap of course videos (5 mins)
3. Overview of Tech Stack (10 mins)
4. Running Python code from Visual Studio Code (30 mins)
5. Type Hints in Python (15 mins)
6. Software Design: Ingenuity and Simplicity (10 mins)

# Slides from the Assigned Videos

# Handcraft

# Knitting

The language used to communicate patterns uses exactly  the same fundamental constructs as Python!!

## Sherbet Stripes

*Notes:* Bright, delicious stripes, vertical on the front and horizontal on the reverse side, make this dishcloth a welcome addition to your kitchen. A simple 4 row repeat of slip stitches creates a fun color work effect that is deceptively simple to work but must be done on double pointed needles to allow you to knit from either end of the work.

**Slip Stitch Pattern (worked over four rows)**

**Row 1 (RS):** With CC, *Sl1 WYIB, k1*, repeat between *'s until 1 st remains, Sl1 WYIB.

**Row 2 (WS):** Slide the work to the other end of the needle and pick up MC to work. *K1, Sl1 WYIB*, repeat between *'s until 2 st remains, K1. Turn work.

**Row 3:** With CC, *Sl1 WYIF, P1*, repeat between *s until 1 st remains, Sl1 WYIF.

**Row 4:** Slide the work to the other end of the needle and pick up MC to work. *P1, Sl1 WYIF*, repeat until 1 st remains, P1. Turn.

### DIRECTIONS

With MC, CO 33 sts.

K 1 row.

Begin Slip Stitch Pattern and work 11 rep of the 4 row rep. (44 rows of patt.)

Break CC yarn.

K 1 row in MC.

BO all sts.

### Finishing

Weave in ends, wash and block to dimensions.

### About the Designer

**Gillian Wynne Grimm** lives in a little white cottage on a tree lined street in Portland, Oregon where she knits, sews and generally enjoys making all manner of crafty and creative things.

Follow along with her adventures at Birchhollowcottage.com.

For pattern support, please contact info@birchhollowcottage.com

# Knitting

# Quantifying the task...

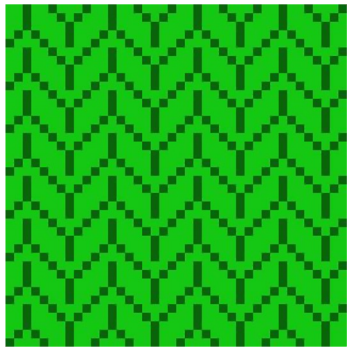
1. If we describe one dimension of a square rag by *n*, how much work is done by the knitter? _____
2. If we have enough yarn for 36000000 stitches, what is the largest rag we could make? _____
3. If each stitch takes a second, what is the largest rag we could make in one evening? _____
4. If it takes an evening to make a 40x40 rag, how long will it take to make an 80x80 rag? _____
5. If it takes time *t* to make an *n* by *n* rag, how long will it take to make a *3n* x *3n* rag? _____

General idea: quantify the size of the problem (*n*) and consider the cost of our task *as that size increases*.

# Quantifying the task...

If we are solving a problem _____ for a given input, we can parameterize the running time of the solution by the size of the input.

We *usually* denote this input size using the variable *n*.



Discussion:

1.

2.

3.

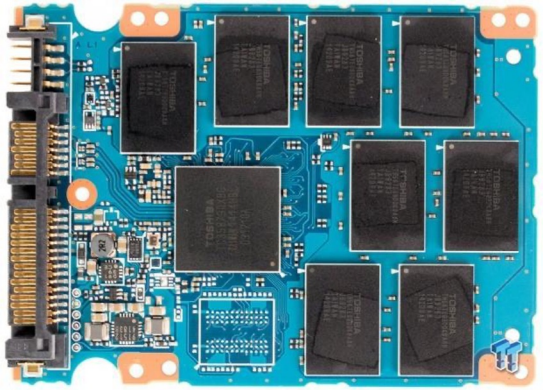
# Quantifying the task...

Suppose we can knit $10^{12}$ stitches per second....

| T(n) \ n | 10 | 100 | 1000 | 10,000 | $10^{12}$ |
|---|---|---|---|---|---|
| $n$ | $10^{-11}$ s | $10^{-10}$ s | $10^{-9}$ s | $10^{-8}$ s | 1 s |
| $n \log n$ | $10^{-11}$ s | $10^{-9}$ s | $10^{-8}$ s | $10^{-7}$ s | 40 s |
| $n^2$ | $10^{-10}$ s | $10^{-8}$ s | $10^{-6}$ s | $10^{-4}$ s | $10^{12}$ s |
| $n^3$ | $10^{-9}$ s | $10^{-6}$ s | $10^{-3}$ s | 1 s | $10^{24}$ s |
| $2^n$ | $10^{-9}$ s | $10^{18}$ s | $10^{289}$ s | | |

The amount of computation we do inside our algorithm actually matters!

# Color - Representation



First, some general questions...

What does data look like to a computer?

Is that enough?

ref: http://arstechnica.com/information-technology/2012/06/inside-the-ssd-revolution-how-solid-state-disks-really-work/2/

# Color - Representation

Can we use bits to represent integers?

| 3 | 1 | 5 | 7 |
|---|---|---|---|
| $10^3$ | $10^2$ | $10^1$ | $10^0$ |

=

| | | | |
|---|---|---|---|
| 3 | 2 | 1 | 0 |

=

=

# Color - Representation

What’s the highest number we can represent with 4 bits?

What’s the highest number we can represent with 5 bits?

How many bits do we need to express 127?

# Color - Representation

Can we use integers to represent colors?

RGB - (red, green, blue), where each “component” is in range 0 through 255.

ex. (255, 0, 127) 

fun calculator for color values: http://colorizer.org

How many bits for 256 values? 3 4 5 6 7
8 9 10

# Color - Representation

RGB - (red, green, blue), where each “component” is in range 0 through 255, with 8 bits for each component.

What color is

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

(stretch) Easier to read if we use “hexadecimal” representation:

Each component is represented by 2 hex digits 0123456789abcdef

ex. #674ea7 

# Overview of the Tech Stack



## Tech Stack

For any computer science (or data science) course, several tools and software packages are needed for this course. These tools are not usually included on your computer, and if they are, some configuration is necessary. This page links out to a guide (depending on your operating system) to install the "Tech Stack" for CPSC 203. A "Tech Stack" is the complete set of tools and technologies needed to accomplish a particular task, in this case, Data Analytics.

As you go through the install guides, remember that perhaps the two most important things you will learn in this course is how to troubleshoot things and achieve familiarity and proficiency with your computer. Be patient, read things carefully, do not be afraid to try things, it's unlikely you will do anything to irreversibly break your computer! Almost everything you do (you're on your own if you decide to take a hammer to your laptop!) can be undone, don't be afraid to ask your peers, TAs, and the instructor on Ed Discussion.

This is an exciting time for you as you begin your Software Development journey!

## Install the Software Stack

Below are links to the software installation guide used for CPSC 203 for the three operating systems that are currently supported.

- macOS
- Windows
- Ubuntu

## Attribution

# Running Python Code in Visual Studio Code

# Type Hints in Python

# Software Design: Ingenuity and Simplicity

# Steve Jobs:

Simplicity is the ultimate sophistication. It takes a lot of hard work to make something simple, to truly understand the underlying challenges and come up with **elegant** solutions. [...] It's not just minimalism or the absence of clutter. It involves digging through the depth of complexity. To be truly simple, you have to go really deep. [...] You have to understand the essence of a product in order to be able to get rid of the parts that are not essential.

# Before Next Class:

Make sure you finish the Tech Stack!