
CSC 120

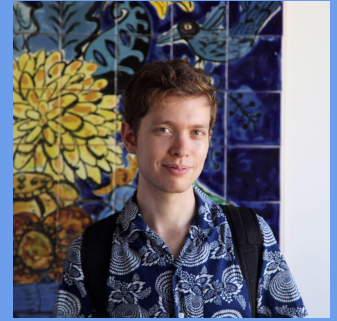
Computer Science for the Sciences

Fall 2018

Mark Kazakevich

csc120-2018-09@cs.toronto.edu

About me: Mark

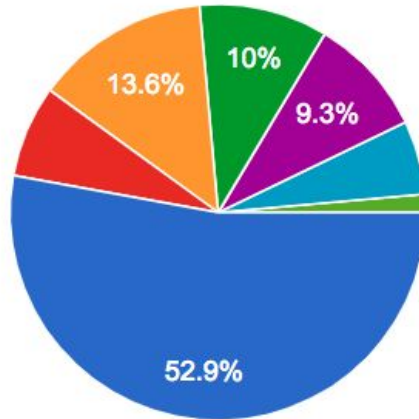


- Studied Computer Science & Physics at UofT
- Involved in a lot of teaching and education work here
 - Instructor in CS department
 - Course Creator in the Department of Physics
 - Created the course **PHY207** - The Physics of Music
- Jazz Pianist by night

About You

From the Welcome survey that many of you filled out:

Which year of study are you currently in?

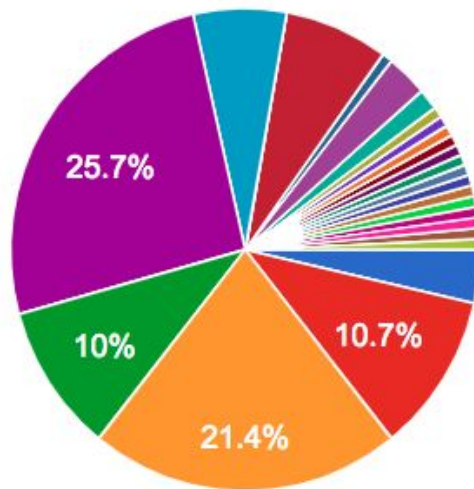


- First year, first term at UofT
- First year, but took a UofT course before
- Second year
- Third year
- Fourth year
- Fifth year+
- Graduate Student
- Other

About You

From the Welcome survey that many of you filled out:

What is your intended major?



- Computer Science
- Mathematics
- Statistics
- Physics
- Life Sciences
- Psychology
- English
- Music

▲ 1/4 ▼

- Economics
- Architecture
- Cognitive Science
- cognitive science
- Cognitive Science
- Rotman Commerce with a Compute...
- actuarial science
- accounting

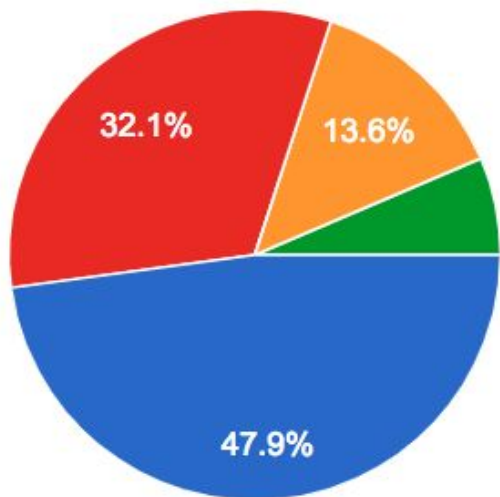
- Ecology and Evolutionary Biology
- Chemistry
- mathematics and english
- Criminology
- Actuarial science
- Cognitive science
- Environmental Science
- commerce

- Compute Science and Cognitive Science
- Actuarial Science
- Religion

About You

From the Welcome survey that many of you filled out:

How would you describe your prior experience with programming?



- I have never programmed before.
- I have written a few lines of code, but don't really know much about programming.
- I have written some small programs before.
- I have more programming experience than described above.

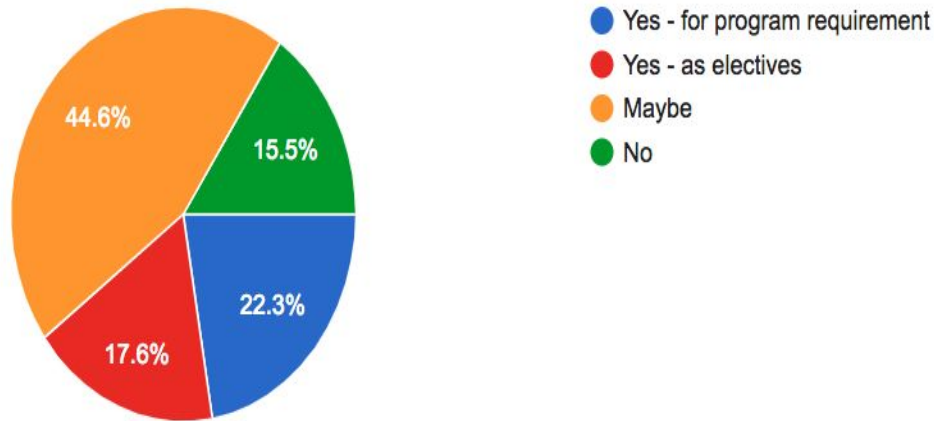
80% of the class has little or no programming experience

About You

From the Welcome survey that many of you filled out:

- Half of students are taking CSC120 as an elective.
- Half of students expecting to take more CSC courses.

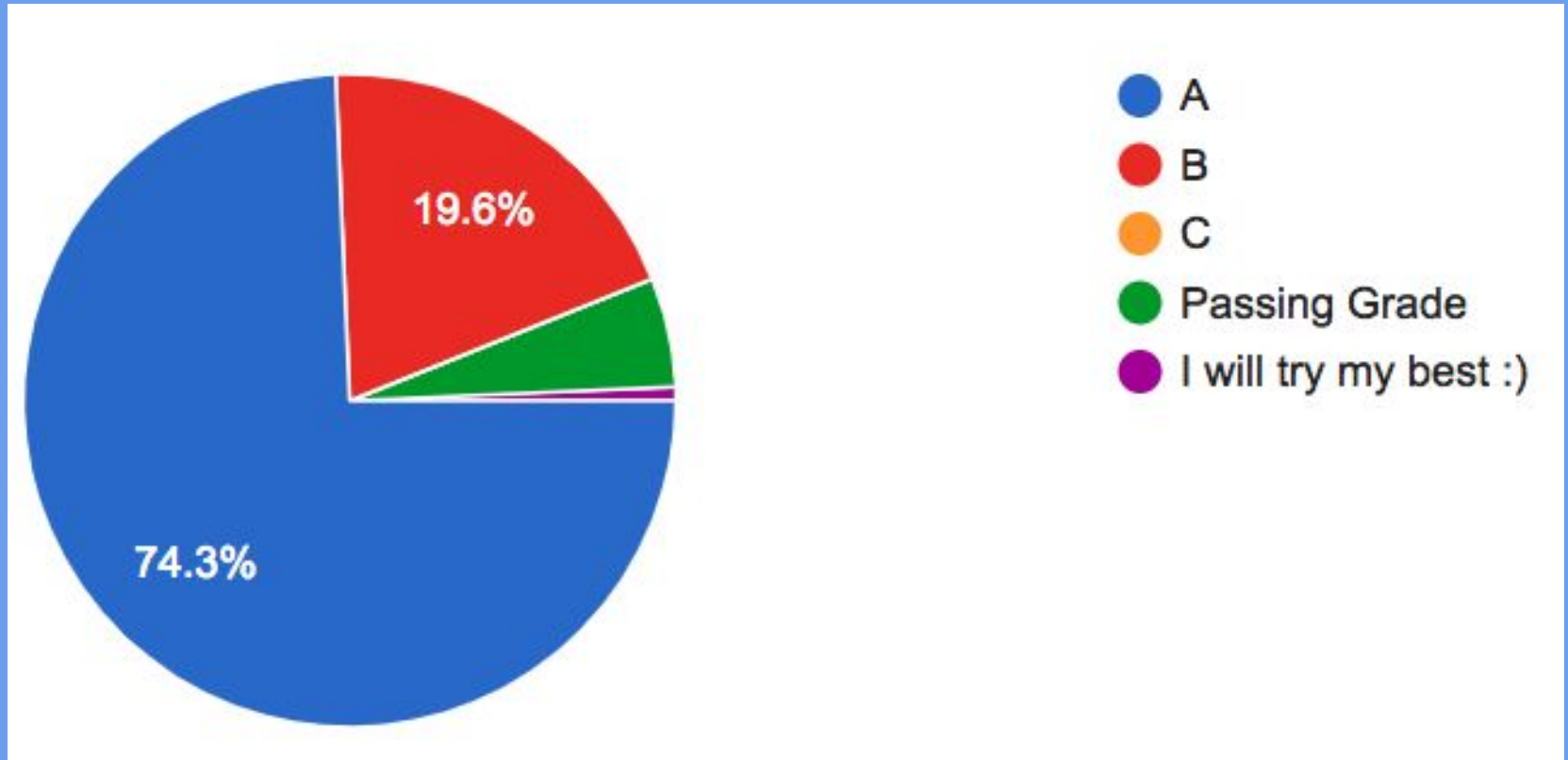
Do you intend to take more computer science courses after CSC120?



Computer Science is very cumulative, important that you build good foundations.

About You

Everyone wants to do well, these slides are about how.



What is CSCI20 all about?

At the end of this course, you will

- know most Python instructions
- be able to take human problems and write Python programs that solve them
- learn to break down complex problems into smaller steps: devising an **algorithm**.
- acquire an appreciation of how computer science **applies** to other disciplines

Who is CSCI20 for?

- Anyone who wants to learn to program!
 - Especially if you want to apply what you learn to another discipline, such as the natural or social sciences
- People who have never programmed before
- People who don't really know what programming is

Computer science for...the Sciences

- **no specific knowledge of any science** is required to take this course.
- The main aim of this course is to teach you how to program, and to give you a lot of opportunities to practise what you learn by writing your own code.

We assume that students in
CSC I 20 have never
programmed before!

Classroom rule: when the instructor poses a question, if you knew the answer before taking this course, do not answer the question.

Do you already know the I20 material?

You may want to **take CSCI48** instead:

Assumes basic Python programming. (strings, lists, sorting, functions, methods)

But if you've programmed in a different language, they have a ramp-up session to get you up to speed on Python

Suitable for those with the equivalent of CSCI08/I20.

Let's talk about our course!

Course Website + Information Sheet

The course website, and the Course Information Sheet, have all the key administrative details.

The course website is here:

<http://www.teach.cs.toronto.edu/~csc120h/fall/>

It will contain the most up-to-date information about the course.

Both the website and info sheet are **required reading**.

(we will go over them a bit)

Lectures

I will explain and demonstrate concepts, mostly by coding live in front of you.

We will also do some in-class exercises

Time/Location

- Mondays, Wednesdays and Fridays in **LM 159** (12-1pm)

Textbook

Optional: Practical Programming (2nd or 3rd edition only)

eBook: (3rd edition)

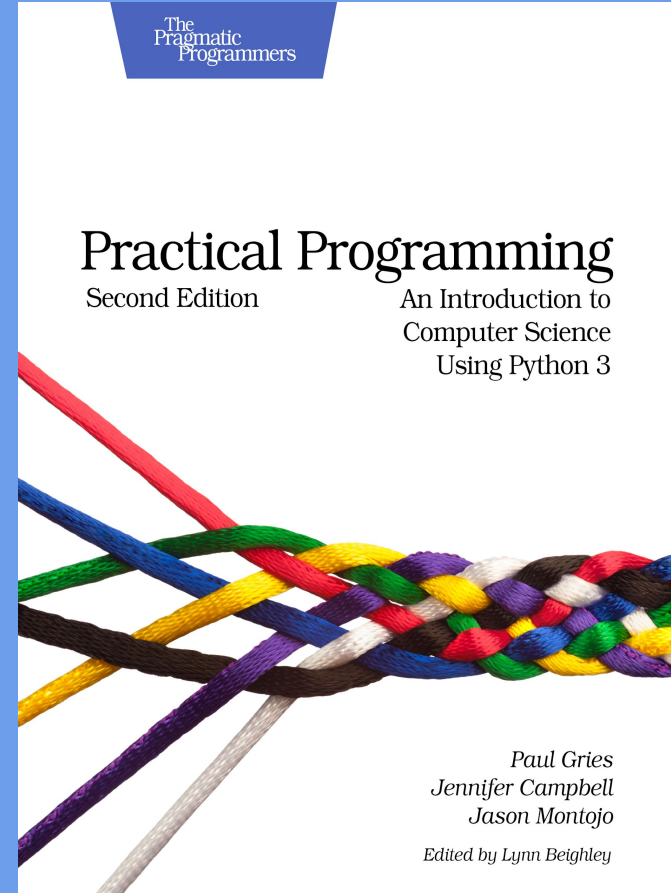
Formats: PDF, ePub, mobi

If you buy the eBook, you can have it right away.

<http://pragprog.com/book/gwpy3/practical-programming>

Paper version: (2nd edition) Sold at UofT bookstore,
amazon.ca, etc.

Everything you need for the course will be on the
website



Coursework Overview

Work	Weight	Comment
Labs (10)	10%	2-hour weekly lab (Thursdays 1-3pm or 3-5pm) 1% each
Exercises (6)	15%	3% each (keep best 5 out of 6)
Assignments (2)	20%	10% each
Midterm	15%	15% During lecture time Think of it as “practice for the final”.
Final Exam	40%	You must get $\geq 40\%$ on the final to pass CSCI20. Otherwise, your final course grade will be no higher than 47.

Labs

When: Thursdays 1:00 – 3:00pm **or** 3:00 – 5:00pm, weekly.

Please sign up for a lab time on ACORN.
We need to know where to put you!

Where: We currently have the following lab rooms reserved:

BA 3175, 3185, 3195

I will let you know which lab room to go to next week.

If you are on the waitlist, come to lab next week and let your TA know.

Marking: 1% per lab, with 10 labs total = 10%

Labs continued

- Labs are to be done in **groups of two students** (pair programming).
- To earn the 1% for a lab, you must arrive on time and work hard.
- When you finish, make sure you show your work to your TA or you may not get credit for the lab.
- The TAs have been instructed not to give credit to students who arrive late or leave early without completing the lab, or who do not try hard.
- **Take advantage of the available help from the TAs!**

Exercises (3% each) and Assignments (10% each)

Handouts will be posted on the course website,
similar to the labs.

You will be submitting these electronically,
using the CS department's submission system,
MarkUs.



Exercises

Will be released after the lab

You'll have about 5 days to do them.

- They aren't meant to be super complicated

Due 11pm on due dates

- No late submissions accepted
- Best 5 out of 6 in case you do miss a due date

Assignments

Assignments are longer and more challenging than the exercises.
You can choose to work with a partner for the assignments.

You'll have a few weeks to them.

- Start early!

Due 11pm on due dates

Late submission policy:

1 hour grace period, then 5% per hour for the next 5 hours, 15% per hour for any additional hour

No other late coursework accepted

No other late assignments will be accepted.

No late exercises will be accepted.

If you can't finish an assignment, you can earn part marks for a good partial solution.

Of course, illness and other emergencies are another matter; contact me as soon as possible if you run into this sort of trouble.

Midterm

During lecture time on **Monday October 22nd**

Will likely be in a different room

Written on paper (like the final exam)



Final Exam

Scheduled by the Faculty of Arts and Science

Exam schedule will be posted here:

<http://www.artsci.utoronto.ca/current/undergraduate/exams>

We can't change it or allow you to write it at a different time!

All exceptions must be handled through Office of the Faculty Registrar: we can't set a makeup test, we can't waive it for you.

You **must get** $\geq 40\%$ on the final to pass CSC120. Otherwise, your final course grade will be no higher than 47.

Doing Your Work

Our labs:

CS Teaching Labs Bahen Centre for Information Technology:

BA3175, BA3185, BA3195, BA3200,
BA2200, BA2210, BA2220, BA2240,
BA2270

You have 24/7 access using your T-card to these rooms (with some exceptions).

Use your Teaching Labs username to log into the lab computers:

See <http://www.teach.cs.toronto.edu/~csc120h/fall/teachinglabs.shtml> for more details.

Working on your own computer

You will need to install Python 3.7 (not 2!) on your own computer.

You can also install Wing IDE 101, the application we'll use to write Python programs.



See the instructions on how to do both on our course website here:

<http://www.teach.cs.toronto.edu/~csc120h/fall/software.shtml>

Laptops

You **do not** need a laptop for this course.

- Can work on a desktop at home or in the Teaching Labs

You can use a laptop in class, but be respectful of those around you

Set up Wing101 sooner rather than later

- The first lab will help with this too

Getting Help

Don't spin your wheels, ask for help!

Instructor Office Hours (more on this in a bit)

120 TA Office Hours (usually in BA 2230)

CSC Help Centre (also BA 2230)

Piazza: An online discussion forum.

The schedule is posted here:

<http://www.teach.cs.toronto.edu/~csc120h/fall/gethelp.shtml>

Mark's Office Hours

2 hours per week

Times TBA

Start week of September 17th, not next week

CSCI20 TA Office Hours

Will be indicated on our calendar on the website.

Should be at least a few each week

CSC Help Centre & First Year Office Hours

CSC Help Centre

When: M-F 2-6pm (starting Sept. 17)

Where: BA2230

Anyone taking a 1st or 2nd year CSC class can go ask questions.

Warning: it gets **busy!**

Discussion Forum: Piazza

Discussion forums are available for you to post questions about the course material.

Ask questions if you're confused!

Provide answers if you know them!

Please don't post solutions (or partial solutions or incorrect solutions)
about any coursework

<http://piazza.com/utoronto.ca/fall2018/csc120h>

Email

Emails to me should be sent to:

csc120-2018-09@cs.toronto.edu

I ask that you do not use any other email to contact me - otherwise your email may get lost

It really, really helps if you start email subjects with “I20: ”

Please read announcements on Quercus before sending email

Use a good subject, such as “I20: issue with partner on A2”

Sign your full name and include your student number and/or UTORid.
(There are hundreds of you and some even have the same name!)

**Let's get serious for a minute about
Academic Offences.**

What *is* an Academic Offence in this course?

- Copying the code of other students
 - Even if it's not an 'exact' copy
- Using code you found online
 - DO NOT copy and paste code you found online that solves any or part of the assignment
- Giving your code to another student

The CS department has software that compares submissions for evidence of similar code.



3 Rules of Thumb

1. All of the work you submit must be your own.
2. You must not submit the work of another student from this term, a student from a previous term, a friend, a tutor, an online source, etc...
3. Your work must not be submitted by someone else.

Tips on how to be good and avoid plagiarism

- **Never look at another assignment solution - this includes going looking for one online!**
- **Never show another student your work.**
 - **Applies even to drafts and to incomplete solutions.**
- **Discuss how to solve an assignment only with the CSC120 TAs, the Help Centre TAs, and the course instructor.**

Good Practices for this course

Lectures

- Come to lectures and participate
- Ask questions => no such thing as a *stupid* question!
- Answer my questions :) See it as a small challenge!

Mark's Office Hours

- Please come! Bring questions or the work you've done.
- "I don't understand topic X" is fine too! :) I am there to explain it again!

Good Practices - cont'd

Practice!

- The effort you put into the class will reflect what you get out of it :)
- Remember: programming is a skill-set! It can be learned. That's what we're here for!
- The material is cumulative. So do something every week.
- Looking at someone else's code (or my solutions in class) and understanding it is **very very very** different than writing the code yourself from scratch.
- Keep a log/journal of common mistakes you might do! Makes you aware of any issues!

Good Practices - cont'd

Take advantage of:

- Piazza Discussion Forum
- CSCI20 Labs and TAs
- CSC Help-Centre Hours / First Year Office Hours

Communicate any issues:

- Use the Anonymous Feedback Form

The key is

To be willing and curious to learn :)

For example, when programming, it's not about:

“Tell me what the error is and fix it for me”

simply a patch; won't hold

but rather:

“Help me learn how I can figure out the error on my code and fix it myself.”

This is learning. We're teaching you a skill-set!

TODO List before next lecture

Check out the course website.

Read the Course Info Sheet.

Log in to Quercus

If you plan on using your own computer, try to install Python and Wing 101

See the Software page on the course website.

If you can't get it installed, we will help you during the first lab next week.

Frequently Asked Questions (FAQ)

Why am I being asked to pay for WingIDE?

There are a few versions of Wing. We are using **Wing101**, which is free. If you've got a message about paying for Wing, you have the wrong version. See the course website Software page for a link to Wing101.

Why does my version of Python look different from yours?

Make sure you are using **Python3**, NOT Python2.

(Python2 is a different programming language, and so if you write your assignments in Python2, we won't be able to run them!)

Any Questions?

See you next week!