APS106

UNIVERSITY OF TORONTO

Fundamentals of Computer Programming

Introduction.

Week 1 | Lecture 1 (1.1)

This Week's Content

- **Lecture 1.1**
  - **Introduction – Why Programming?**
- **Lecture 1.2**
  - Computers, Programs, and The Coding Toolbox
- **Lecture 1.3**
  - Variables, Expressions, and Operators

Ben

- Bachelor of Electrical & Biomedical Engineering @ McMaster University 2015
  - Realized the power of programming (and how it can be used for good, not evil)
- Master of Biomedical Engineering @ University of Toronto 2020
  - Hold up… people will pay me for ‘playing’ on my computer?
- In between obtaining degrees I spent time programming a few different projects…

Projects for children with autism spectrum disorder

Listening...

Hello
Hey
Good afternoon

Hi!

Good Vibes

A wheelchair vibration measurement tool

9:55 AM Mon Jan 3
100%
Good Vibes
Home
Live Table
Seating Pad Session
Backrest Session
Seating Pad Session
Z (Up/Down) Vibration
Status
Z (Up/Down) Acceleration
Session RMS
1 Hour Caution Zone
Session Caution Zone
1 Hour Health Risk Zone
Session Health Risk Zone
Acceleration (m/s²)
4
3
2
1
0
-1
0
1
2
3
4
5
6
7
8
9
10
Time (seconds)
Start/Stop
Summary
Save Data

More about Ben…

Being Respectful is an Expectation

- Talking
- Working on other courses
- Eating
- Being Late for Lectures

Being Respectful is an Expectation

- Talking
- Working on other courses
- Eating
- Being Late for Lectures

Ask Questions!

- Ask questions as much as possible.
- Ask **as soon as** you feel confused!
- Ask each other questions in-person or on Piazza and help if you think you know the answer
- Never forget - if you have a question, there's always other people in the room wondering the same thing. They will be thankful you asked it!

Course Information

- **Quercus:**
  - APS106H1 S LEC0101 (Website contains all information).
- **Textbook:**
  - ***Programming in Python 3***.
  - A link to the UofT bookstore where you can purchase the textbook can be found on the Syllabus page.
- **Piazza:**
  - *https://piazza.com/utoronto.ca/winter2025/aps106/home*

Grading

Petitions: First-Year Office.

Reflections (13 in total)	total: 2%
Labs (10 in total, 1% each)	total: 10%
Term Test 1 (Thurs Feb 13)	total: 15%
Term Test 2 (Tues March 18)	total: 23%
Final Exam	total: 50%

APS106 Golden Tip for Success

Complete weekly modules DURING that week

Week 1 | January 6 - 10

- Lecture 1.1 - Introduction to APS106
- Lecture 1.2 - Computers, Programs, and The Coding Toolbox
- Lecture 1.3 - Variables, Expressions, and Operators
- Tutorial 0 - Setting Up A Development Environment
- Lab 0
  Jan 17 10 pts
- Reflection 1
  Jan 13 1 pts
- Textbook Readings 1
- How Does That Work? (HDTW)

Lecture Schedule

Section	Monday	Tuesday	Wednesday	Thursday	Friday
LEC0102		1:00 pm – 2:00 pm MS2170	1:00 pm - 2:00 pm SF1101		1:00 pm - 2:00 pm SF1101

- Lectures will be in person in **MS 2170 & SF 1101**.
- Lecture content can be found on Quercus using the Kinsella & Goodfellow links.

Tutorial Schedule

One online tutorial TBD.

Week	Tut Content	Monday	Tuesday	Wednesday	Thursday	Friday
Week 1 (Jan 6 - Jan 10)	**Tutorial 0**		**TUT0102** 2:00 PM - 3:00 PM **TUT0105** 5:00 PM - 6:00 PM		**TUT0106** 12:00 PM - 1:00 PM **TUT0101** 3:00 PM - 4:00 PM **TUT0104** 3:00 PM - 4:00 PM **TUT0108** 5:00 PM - 6:00 PM	**TUT0103** 3:00 PM - 4:00 PM **TUT0107** 3:00 PM - 4:00 PM
Week 2 (Jan 13 - Jan 17)	**Tutorial 1**		**TUT0102** 2:00 PM - 3:00 PM **TUT0105** 5:00 PM - 6:00 PM		**TUT0106** 12:00 PM - 1:00 PM **TUT0101** 3:00 PM - 4:00 PM **TUT0104** 3:00 PM - 4:00 PM **TUT0108** 5:00 PM - 6:00 PM	**TUT0103** 3:00 PM - 4:00 PM **TUT0107** 3:00 PM - 4:00 PM

Lab Schedule

Week	Monday	Tuesday	Wednesday	Thursday	Friday
Week 1 (Jan 8 - Jan 12)				**Lab 0 Released** 6:00 PM	**PRA0108** 3:00 PM - 5:00 PM
Week 2 (Jan 15 - Jan 19)	**PRA0107** 10:00 AM - 12:00 PM **PRA0104** 12:00 PM - 2:00 PM **PRA0101** 1:00 PM – 3:00 PM	**PRA0106** 4:00 PM - 6:00 PM **PRA0111** 4:00 PM - 6:00 PM	**PRA0109** 3:00 PM - 5:00 PM **PRA0110** 4:00 PM - 5:00 PM	**PRA0102** 10:00 AM - 12:00 PM **PRA0103** 10:00 AM - 12:00 PM **PRA0105** 4:00 PM – 6:00 PM **Lab 1 Released** 6:00 PM	**PRA0108** 3:00 PM - 5:00 PM **Lab 0 Due** 11:59 PM

Reflections

- There are 13 reflections throughout the term.
- Reflections are worth a total of 2% of your final grade.
- Full marks for successful completion of all reflections.
- Reflections are used to take the pulse of students in the course.
- Insights from reflections are used to iteratively improve the course week-by-week.
- Reflections are there to help you, so please answer honestly.

Reflections

- Reflection include question like:
  - Did you attend lecture this week? If not, why?
  - Do you understand the topics covered this week?
  - Did you attend tutorial this week?
  - Which lecture section did you attend?
  - What will this code output?
  - Write one word to describe how you feel about the course.

Reflections Schedule

Week	Reflection	Monday	Tuesday	Wednesday	Thursday	Friday
Week 1 (Jan 6 - Jan 10)	**Reflection 1**					**Reflection 1** Released 6:00 pm
Week 2 (Jan 13 - Jan 17)	**Reflection 2**	**Reflection 1** Due 1:00 am				**Reflection 2** Released 6:00 pm
Week 3 (Jan 20 - Jan 24)	**Reflection 3**	**Reflection 2** Due 1:00 am				**Reflection 3** Released 6:00 pm
Week 3 (Jan 27 – Jan 31)	**Reflection 4**	**Reflection 3** Due 1:00 am				**Reflection 4** Released 6:00 pm

Tips for Success

- Do not wait until the last minute.
- Programming is a Language.
- Work Together (except during the test/exam)
- Work Smart (effort doesn't necessarily equal progress).
- Time Management.
- Know when (<u>and how</u>) to ask for help.

Tips for Success

- **88%** of your final grade is based on two term tests and an exam.
- For these tests and exams, you'll be asked to write code by hand.
- You will not be able to achieve a high mark by reading the textbook and lecture notes. You need to PRACTICE, PRACTICE, PRACTICE!
- How can you practice writing code?
  - **Lecture Breakout Sessions.**
  - **Labs (think of this as studying).**
  - **Practice Problems.**

Writing A Lot Of Code During The Term

=

Higher Grade

What is this course all about?

- Learn the **fundamentals** of programming.
- **Programming as a tool** for all engineering disciplines.
- **Enjoy coding** as a new and meaningful skill.
- **Experience engineering design** through programming.

Why Programming?

StartUp HERE

Why Programming?

- Computers are everywhere:
  - Learn to interact with them.
  - Simplifies our jobs.
- Entrepreneurial opportunities ($$$).
- Necessary for engineering jobs.
- Research and development.
- Because it's fun!

#DISRUPTMINING

PRESENTED BY GOLDCORP — CO-HOSTED BY KPMG

KORE GEOSYSTEMS

$1,000,000 Investment.

What is Programming?

- A way of telling a computer what to do.
- A computer can't infer (…yet).
  - Need to tell a computer every single step it needs to do, in a language it can understand.
  - How would you request an egg for breakfast to a chef and to a computer/robot?

- **To a Chef**
  1. Sunny-side up, please!

- **To a Computer**
  1. "Turn on stove"
  2. "Take out pan"
  3. "Take one egg out of fridge"
  4. "Crack egg"
  5. "Pour egg into pan"
  6. "Wait 5 minutes"

How to Program a Computer.

Programmer

Compiler

Assembler

English

Pseudocode

Programming Language

Assembly Code

Machine Code

Why Python?

- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses Python.

C,
C++, JAVA

PYTHON

Why Python?

- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses Python.

Java

Lower-Level

```
public class HelloWorld
{
    public static void main(String[ ] args)
    {
        system.out.println("Hello World!");
    }
}
```

python

Higher-Level

```
print("Hello World!")
```

Why Python?

- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses Python.

Increasing Program Size (# lines of code)

Matlab, SQL

C++, Java, Python

C, Fortran, Pascal

Assembly Code

Machine Code

Increasing Ease-of-Use

Why Python?

- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses Python.

python

```
if x > 10:
    print("x is greater than 10")
```

You don't know Python.
Can you figure out what this code does?

Why Python?

- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses Python.

Why Python?

- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses Python.

Why Python?

- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses Python.

PyTorch

LANDING AI

OpenAI

Azure Machine Learning

XGBoost

TensorFlow

scikit learn

Why Engineers?

- Ok, so we've explained why coding is important and why we're learning to code in Python.
- **BUT**, you're all different types of engineers, right?
  - Mechanical
  - Materials
  - Civil
  - Chemical
  - Mineral

Why Engineers?

- Again, why do engineers need to know how to code?
- In the 20th century, engineers didn't need to know how to code.
- In the 21th century, coding is an **essential skill** and any engineer without solid coding abilities will be at a major disadvantage.

Why Engineers?

- Five technology trends have changed the game for engineers.

The Cloud

Cheap Sensors

Big Data

Artificial Intelligence

The Internet Of Things

Mechanical Engineers

Realtime Diagnostics & Tuning

CNET

Tech Money Home Wellness Home Internet

Cars > Auto Tech

Tesla can tune your Model S or X's suspension via software update

Materials Engineers

Digital Image Analysis of Material Microstructure

Civil Engineers

Urban Transportation Disrupted

Mineral Engineers

Semi-Autonomous Mines

Chemical Engineers

Synthesizing new compounds

Sci-fi Inspiration

CompBioMed

Simulation of a potential anti-cancer drug binding to HDAC8 enzyme (linked with acute myeloid leukemia)

Programming Skills Are In Demand

- According to the *Developer Survey by StackOverflow*, Python was one of the most in-demand technologies of 2018, 2019, 2020, and 2021.
- As of 2021, it was ranked the most wanted programming language by professional software developers.

Python Skills Are In Demand

- According to the *TIOBE Index*, Python overtook C and Java in 2022 as the world's most popular programming language

Python Skills Are In Demand

- Since 2024, Python has become the significantly most popular language on the planet!
- *TIOBE* attributes this to Python's data mining and AI libraries

One of the reasons for choosing Engineering!

- Finally! A course that isn't Chem, Bio, Physics, or Math
- This (as well as your engineering design course) is what separates you from many other sciences
  - And makes you **MUCH** more valuable….

Before Next Class:

- How Does That Work?
  - Modules -> Course Resources -> How Does That Work? (HDTW)

How Does That Work? (HDTW)

How Does That Work (HDTW) is a series of short instructional videos focused on the tools we use in this course.

HDTW - Installing Python & Anaconda

HDTW - Anaconda Navigator

HDTW - Jupyter Notebooks

HDTW - UofT JupyterHub

HDTW - IDEs & VSCode

HDTW - Debugging (VSCode)

HDTW - Environments & Installing Packages

  - In Tutorial 0 this week -> Set up your development environment.

APS106

UNIVERSITY OF TORONTO

Introduction.

Week 1 | Lecture 1 (1.1)