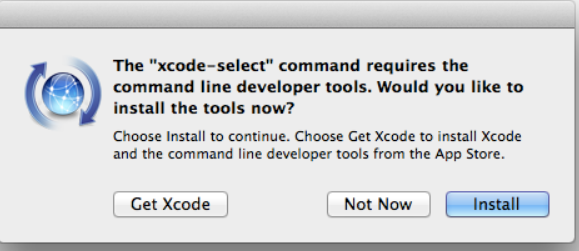
00 What are Data Structures?

## What are Data Structures?

1. Data Structures in a Nutshell
   1. They are the containers which our data is stored in
   2. They are NOT databases
   3. Think of them as different ways of writing your cod
2. Why use them?
   1. Provides the basis of algorithms as a whole
   2. Regardless of which field you want to be in (AI, game dev, DevOps, etc.), you will need to know what data structures are so that you can not only build software but also correct them.

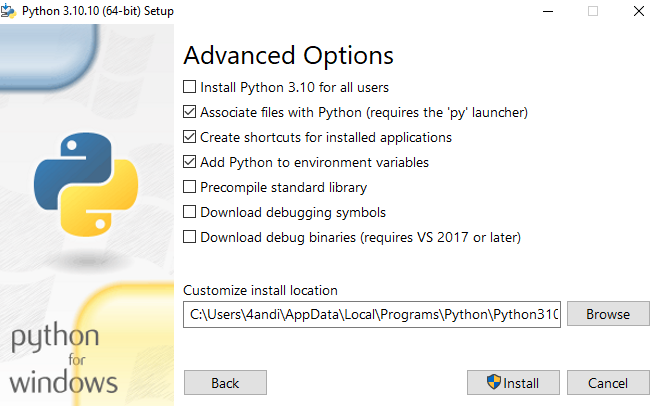
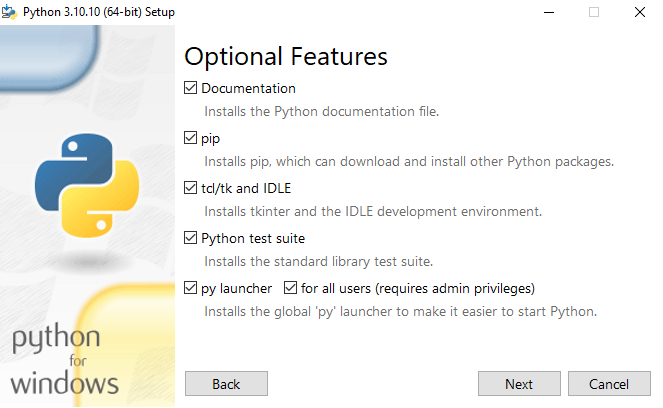
## How to set up your computer for this class

1. Make sure your computer contains the GCC/GPP GNU C++/C compilers so that it can run the code
2. If you’re planning on using C++ on WINDOWS,
   1. If it doesn’t, here are the steps to install [WINDOWS]
      1. Download [MSYS2](https://github.com/msys2/msys2-installer/releases/download/2024-01-13/msys2-x86_64-20240113.exe) (or MinGW-64) which is what enables your computer to set up a development environment to build and compile programs on Windows
      2. Run the Installation Wizard
      3. Once complete, make sure the RUN MSYS2 now box is checked and select Finish.
      4. A terminal window will appear so you must enter this command  
         pacman -S --needed base-devel mingw-w64-ucrt-x86\_64-toolchain
      5. Accept everything by pressed ENTER
      6. Enter Y to proceed with Installation
      7. Add the MinGW-w64 bin folder to the Path Environmental Variable.
      8. To Check if the MinGW installation has worked, enter these commands in a completely new terminal   
         gcc --versiong++ --versiongdb --version  
         g++ --version  
         gdb --version
      9. If it all goes to plan, it should be working.
3. If you’re planning on using C++ on MAC,
   1. Open the Terminal Window
   2. Enter the command xcode-select —install and press enter



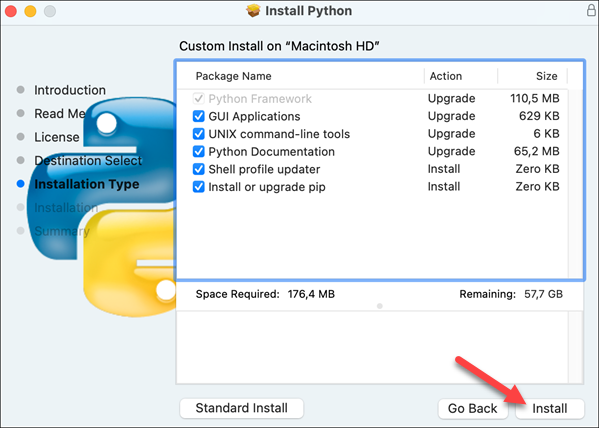
* 1. Click install
  2. Then click done
  3. Type clang —version to make sure that it works

1. Here are the steps to install C++ on any UNIX/LINUX systems
   * 1. Open the Terminal
     2. First type in this command to update the sytem  
        sudo apt-get update
     3. Enter your password
     4. Finally type in this command to install g++  
        sudo apt-get install g++
     5. This will install g++ for UBUNTU
     6. For fedora systems, it’s  
        sudo dnf install g++
2. If you’re planning on using Python,
   1. Go to the python.org website
   2. Click download on the latest release
   3. Follow the instructions on the executable
   4. Click everything



* 1. Now if you want to add to the Environmental Variables, Go to Start and enter advanced system settings in the search bar.
  2. Click View advanced system settings.
  3. In the System Properties dialog, click the Advanced tab and then click Environment Variables.
  4. Depending on your installation:
  5. If you selected Install for all users during installation, select Path from the list of System Variables and click Edit.
  6. If you didn’t select Install for all users during installation, select Path from the list of User Variables and click Edit.
  7. Click New and enter the Python directory path, then click OK until all the dialogs are closed.

1. How to install Python to MAC
   1. It usually comes pre-isntalled with MAC
   2. But if you don’t have it
   3. Go to python.org and download Python.
   4. Run installer



* 1. Afterwards verify

1. To install on UNIX/LINUX
   1. Usually comes pre-installed
2. Honestly, I’m only going to cover C++ and python because both are the most common languages used for Programming nowadays.

## What sort of Data Structures will we be looking at?

1. TODO: (collectively) Write a brief description of what each one is. Just 1-2 sentences. Don’t complicate it.
2. Arrays
   1. A linear, single-direction container that primarily stores data in the form of integers (int), floats (float), doubles (double), and characters (char)
3. List
   1. A python version of Arrays but much more flexible
4. Hash Stables
5. Vectors
6. Trees
7. Graphs
8. Lists
9. Stacks
10. Queues

## Pointers and references

1. TODO: write what are pointers and references
2. TODO: Write what exactly is memory management and why is this so important in C++ and data structures

## Dynamic vs Static Memory

1. TODO: write what is the difference between the two
2. TODO: elaborate more on memory managment

## Big-O Notation: How we measure how quickly a code compiles

1. VERY IMPORTANT
2. TODO: Write what is BIG-O notation
3. TODO: Write list of diferent BIG-O Notations
4. TODO: Every single Big-O notation
   1. TODO: 1
      1. Each one also includes a handy-dandy graph illustrating what the compilation times look like
   2. TODO: 2
   3. TODO: 3
   4. TODO: 4
   5. TODO: 5

# OUTLINE DONE