Lab 2: Setup, Machine Model, Testing, Version Control

Due **Sun Sep 17th, 2023 at 8:00PM ET** on Gradescope. Each lab is graded for attendance (individual) and completion (group). ***Note: Don't forget to add your pod members to the Gradescope submission!**

One group member should create an editable copy of this worksheet to fill out and share the document with all other group members. (*File > Make a Copy* in the menu.) Please do not change the formatting of the document or which exercises appear on which slides. We use the placement of questions on each page to assist with grading.

Each group member is assigned a letter (A/B/C/D), based on the **lab logistics email** you received. Fill in the table below. Your **letter** will be **the same** throughout the term. The **roles** assigned to each **letter/person** rotate each week. We'll explain the roles in more detail below.

	Name	**Role**
Person A	Abhinav	Leader
Person B	Zichen	Scribe
Person C	Gage	Coder I
Person D	Michael	Coder II

The "person letters" are used throughout the lab document to help structure exercises and split up responsibility for leading discussion. If you briefly skim through the document, you'll see tables with "Person A", "Person B", etc. It will be nice to change these to your actual names. The ***Scribe*** should do a find/replace (select *Edit > Find and replace* from the menu) to change all instances of "Person A" to that person's name, and so on.

If your group does not have exactly four members, please work out a strategy for sharing the roles that work for you.

We expect this worksheet will take about 60 minutes to complete.

You are expected to lead the meeting yourselves and collaboratively work through the exercises, but your staff members are here if you need them. If you have questions, please let your staff know!

Setup Tutorial Check-In

For the projects in particular, it's important to get set up for C++ development and feel comfortable with your programming environment.

The "Setup Tutorial" linked from eecs280.org got you started on this for project 1. It's OK if it takes a while before the setup and your environment feels natural. Sometimes getting everything configured can be half the battle! Each member should share some information about their development environment and how their setup has been going.

	Windows, Mac, or Linux?	Which IDE? (e.g. Visual Studio, xcode, VS Code)	How are you feeling about each component of your "setup" so far? Use a 1 to 5 scale. 1 = halp! 3 = ok 5 = great!		
			Using the terminal	IDE and editing code	Debugger
Person C	Windows	VS Code	2	5	1
Person D	Mac	VS code	2	4	2
Person A	Mac	Visual Studio	3	5	3
Person B	Windows	VS Code	2	5	3

Brainstorm two "setup-related" tips or tricks you either learned from the setup tutorial or elsewhere that have been helpful as you're getting started in 280.

1.It's good to turn on the sanitizer manually every time in the Makefile	2.Rename all of your files, it makes it easier to find what you need.

Identify two challenges or questions you have about setup. At some point during your lab, make sure to check in with your lab staff on these.

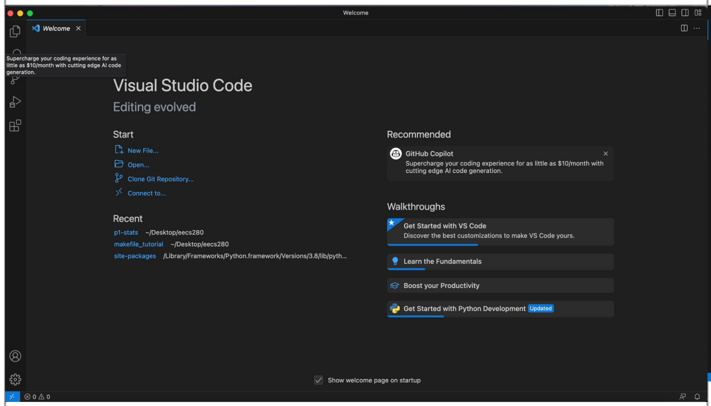
1. Have trouble using the WSL terminal (a.k.a linux system)
2. Using the terminal to set up projects.

Setup Tutorial Check-In

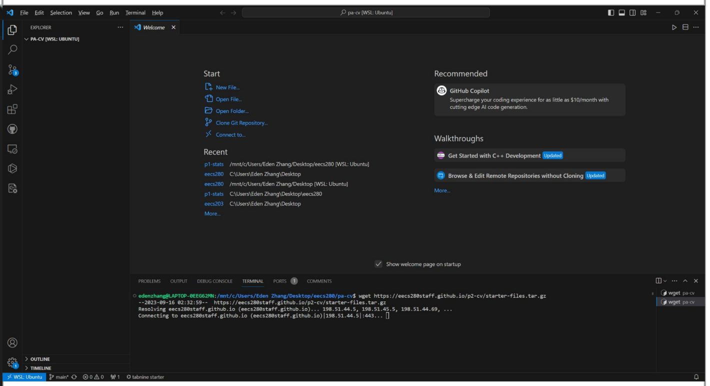
Each group member should paste a screenshot of their development environment (e.g. Xcode, VS Code, Visual Studio). Make sure not to include your project solution code. :)

If you don't have 4 group members, you may write "None" in the corresponding box.

Person A

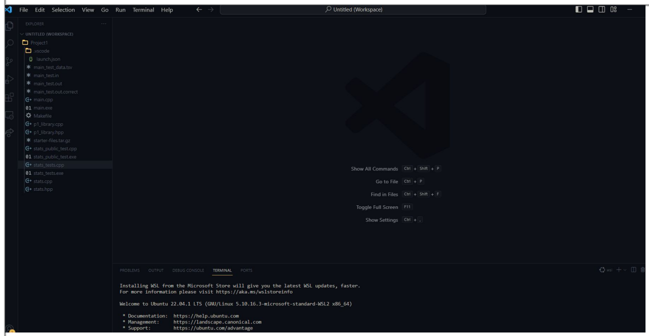

Welcome
Visual Studio Code
Editing evolved
Start
New File...
Open...
Clone Git Repository...
Connect to...
Recent
p1-stats ~/Desktop/eecs280
makefile_tutorial ~/Desktop/eecs280
site-packages /Library/Frameworks/Python.framework/Versions/3.8/lib/pyth...
Recommended
GitHub Copilot
Supercharge your coding experience for as little as $10/month with cutting edge AI code generation.
Walkthroughs
Get Started with VS Code
Discover the best customizations to make VS Code yours.
Learn the Fundamentals
Boost your Productivity
Get Started with Python Development Updated
Show welcome page on startup

Person B


File Edit Selection View Go Run Terminal Help
pa-cv [WSL: Ubuntu]
EXPLORER
PA-CV [WSL: UBUNTU]
Welcome
Start
New File...
Open File...
Open Folder...
Clone Git Repository...
Connect to...
Recent
p1-stats /mnt/c/Users/Eden Zhang/Desktop/eecs280 [WSL: Ubuntu]
eecs280 C:\Users\Eden Zhang\Desktop
eecs280 /mnt/c/Users/Eden Zhang/Desktop [WSL: Ubuntu]
p1-stats C:\Users\Eden Zhang\Desktop\eecs280
eecs203 C:\Users\Eden Zhang\Desktop
More...
Recommended
GitHub Copilot
Supercharge your coding experience for as little as $10/month with cutting edge AI code generation.
Walkthroughs
Get Started with C++ Development Updated
Browse & Edit Remote Repositories without Cloning Updated
More...
Show welcome page on startup
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1 COMMENTS
wget pa-cv
wget pa-cv
OUTLINE
TIMELINE
WSL: Ubuntu
main*
tabnine starter

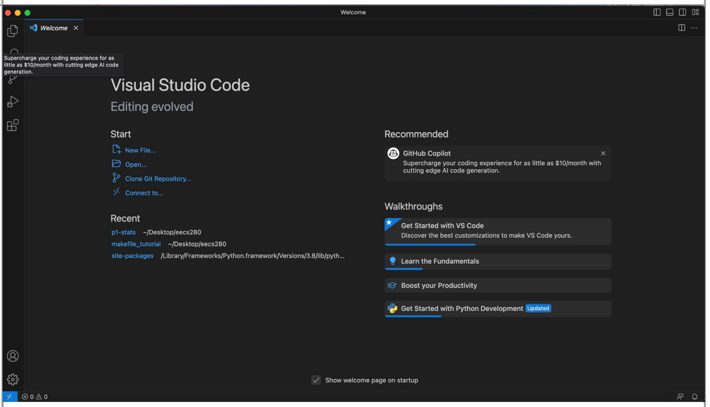
```
edenzhang@LAPTOP-0EEG62MN:/mnt/c/Users/Eden Zhang/Desktop/eecs280/pa-cv$ wget https://eecs280staff.github.io/p2-cv/starter-files.tar.gz
--2023-09-16 02:32:59--  https://eecs280staff.github.io/p2-cv/starter-files.tar.gz
Resolving eecs280staff.github.io (eecs280staff.github.io)... 198.51.44.5, 198.51.45.5, 198.51.44.69, ...
Connecting to eecs280staff.github.io (eecs280staff.github.io)|198.51.44.5|:443... 
```

Person C


File Edit Selection View Go Run Terminal Help
Untitled (Workspace)
EXPLORER
UNTITLED (WORKSPACE)
Project1
Show All Commands Ctrl + Shift + P
Go to File Ctrl + P
Find in Files Ctrl + Shift + F
Toggle Full Screen F11
Show Settings Ctrl + ,
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
wsl

```
Installing WSL from the Microsoft Store will give you the latest WSL updates, faster.
For more information please visit https://aka.ms/wslstoreinfo

Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.10.16.3-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
```

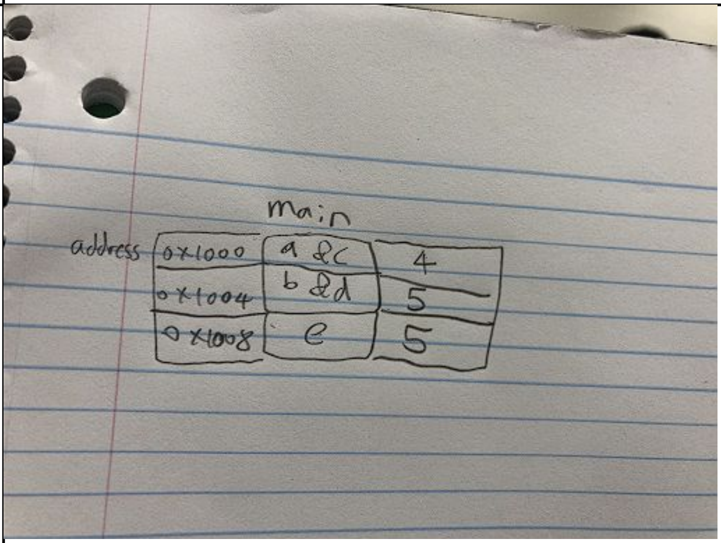
Person D


Welcome
Visual Studio Code
Editing evolved
Start
New File...
Open...
Clone Git Repository...
Connect to...
Recent
p1-stats ~/Desktop/eecs280
makefile_tutorial ~/Desktop/eecs280
site-packages /Library/Frameworks/Python.framework/Versions/3.8/lib/pyth...
Recommended
GitHub Copilot
Supercharge your coding experience for as little as $10/month with cutting edge AI code generation.
Walkthroughs
Get Started with VS Code
Discover the best customizations to make VS Code yours.
Learn the Fundamentals
Boost your Productivity
Get Started with Python Development Updated
Show welcome page on startup

Value Semantics and References

Draw a memory diagram of the following program (which you will use to answer some questions below). ***Coder I*** should draft the drawing (e.g. using the zoom whiteboard, a Google Jamboard, a note-taking program, or even drawing directly on the slides). Paste a screenshot of your drawing into the box below.

Code	**Diagram**

`int main() {`
`   int a = 4;`
`   int b = 5;`
`   int &c = a;`
`   int &d = c;`
`   int e = d;`
 	
`   // assignment statements go here`
`}`
	 main
	address
	0x1000
	0x1004
	0x1008
	a &c
	b &d
	e
	4
	5
	5

Answer the following questions about the code and its behavior at runtime.

Question	**Discussion Leader**	**Answer**
How many variables are there?	Person D	3
How many objects will there be?	Person A	5
What is the <u>minimum</u> number of assignment statements needed to set the value of every variable to 0 (assuming no one-line, multiple assignments like `a = b = 0;`)?	Person C	2
Is the diagram you drew from the source code or the run time perspective?	Person B	Run time perspective

Coding Exercise: *k*-th Highest Value

In this section, you will implement a function similar to what you'll find in Project 1.

```
// REQUIRES: v.size() > k, k >= 0
// MODIFIES: nothing
// EFFECTS:  Returns the k-th highest value in v.
//           Note that k is 0-indexed, so k == 0 is the highest value
//           Example: v == { 6, 2, 3, 6 }, k == 2, returns 3
int kth_highest(std::vector<int> v, int k);
```

Before coding, it's useful to think of several examples of how the function should behave. You can reuse these examples as tests later. Everyone should brainstorm at least one additional test with example inputs for v and k as well as the corresponding desired output.

Discussion Leader	**v**	***k***	**Return Output**	**What kind of test case? (e.g. general, edge case)**
Example	`{ 6, 2, 3, 6 }`	2	3	general
Person D	`{-1, 1, 2, 4}`	3	-1	Edge case
Person A	`{ 5, 1, 4, 2 }`	2	4	general
Person B	`{5, 5, 5, 5}`	3	5	Edge case
Person C	`{1,1,1,1,1,1,1,2,3,4}`	2	2	general

Write code for **one** of the tests below, using `assert` to verify that a call to `kth_highest` with the test inputs actually produces the correct output. Refer to the test examples given `stats_tests.cpp` in the project 1 starter code. **Coder II** should write the code.

```
void test_kth_highest() {
      // your test code here!
    vector<int> v {5, 5, 5, 5};
    assert(kth_highest(v, 3) == 5);
}
```

Before writing code, it's also useful to know what strategy your code will use to solve the problem. There are many ways to solve this problem - one method is provided for you. Brainstorm two alternate strategies as a pod. It may help to take another look at `p1_library.hpp`, which is included as an appendix at the end of this worksheet.

Discussion Leader	**Strategy**	**+ One reason why the strategy is good** **+ One reason why the strategy is not good**
Person C	Find the maximum value in v. Set all the entries in v that are equal to that maximum value to -2147483648, the lowest possible int. Repeat this `k-1` times. Then return the maximum value in the array.	+it works -it is a bit silly
Person A	Sort the vector from lowest to highest Go through the vector with a for loop and the element would be i - k.	+it works -easier to flip the chronological order
Person B	Sort the vector v from the highest to the lowest, then access the k element in the vector.	+light coding -requires knowledge in sorting strategies in C++

As a pod, select one of these strategies to implement. Why did you choose this one?

Discussion Leader	**Which strategy did you choose and why?**
Person D	Sort the vector v from the highest to the lowest, then access the k-1 element in the vector. We chose this because it’s more efficient.

Write the code for `kth_highest` below. **Coder I** is responsible for typing the code, but everyone is expected to contribute and work together!

Your code here

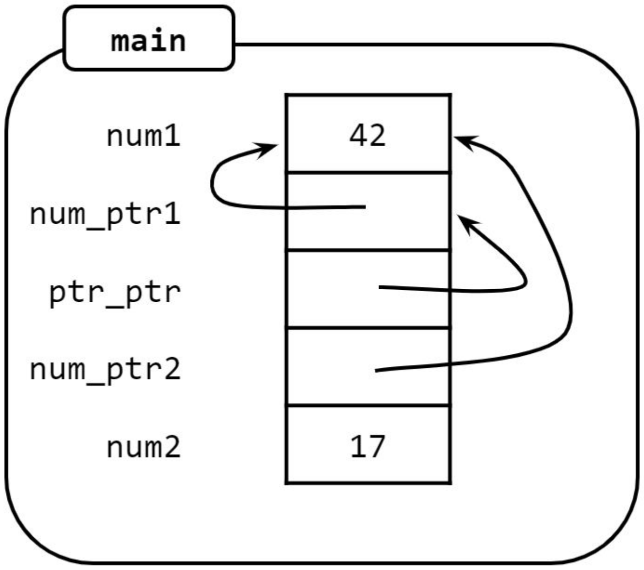
```
int kth_highest(std::vector<int> v, int k) {
  for(size_t i = 0; i < v.size(); ++i) {
     for(size_t j = i; j < v.size(); ++j) {
         if(v.at(i)<v.at(j)){
            Int temp = v.at(i);
            v.at(i) = v.at(j);
            v.at(j) = temp;
         }
      }
   }

  return v.at(k);
}
```

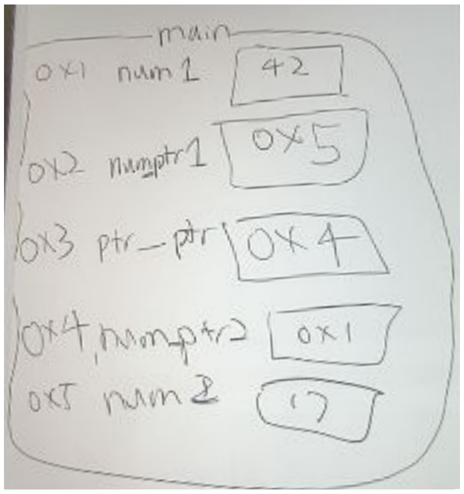
Look back at the example inputs and outputs you made earlier. The ***Leader*** is responsible for talking through what the code will do with one of those sample inputs.

Memory diagrams with pointers

Consider the following code and memory diagram. The diagram is intended to represent the state of memory at the breakpoint.

```
int main(){
  int num1 = 42;
  int* num_ptr1 = &num1;
  int** ptr_ptr = &num_ptr1;
  int* num_ptr2 = num_ptr1;
  int num2 = 17;
  *ptr_ptr = &num2;
  ptr_ptr = &num_ptr2;
  // Breakpoint
}
```



Question	Discussion leader	Your answer
How is the memory diagram incorrect?	Person C	Most boxes shouldn't touch cuz they are not arrays, num_ptr1 should point t num2, ptr_ptr should point at num_ptr2
Draw a corrected version of the memory diagram and insert a picture/screenshot.	Person D	 main 0x1 num1 42 0x2 numptr1 0x5 0x3 ptr_ptr 0x4 0x4 numptr2 0x1 0x5 num2 17

Functions and Pass-by-Pointer

Consider the following code:

```
int func(int *ptr) {
  ++(*ptr);

  int z = *ptr;
  ++(*ptr);
  ++ptr;
  return z;
}

int main() {
  int x = 1;
  int y = 2;
  int z = 3;
  z = func(&x);
}
```

Take time **individually** to trace through the code, drawing a memory diagram as you work. When individual members are finished, the ***leader*** should call the group together to resume discussion and ask others to answer these questions based on their memory diagrams.

Question	Your answer
Can the `func` function modify the value of the variable x? Why or why not?	Yes because it has the location of the object x
Does the line `++ptr;` do anything interesting? Why or why not?	Increments the address the pointer is pointed at by one r bytes
What is the final value of x just before `main()` returns?	`X = 3`
What is the final value of y just before `main()` returns?	`Y = 2`
What is the final value of z just before `main()` returns?	`Z = 2`

Unit Testing Framework Tutorial

In project 1, you wrote unit tests to check for expected behavior using `assert`.

However, this approach can be a bit clunky - the very first failed `assert` crashes out of the program and the rest of your tests don't even run.

Starting with project 2 and beyond, you'll use a **unit testing framework** instead, which provides special testing assertions (e.g. `ASSERT_EQUAL`). Instead of crashing on failure, the framework records each test success/failure and reports overall results at the end.

As a group, work through our Unit Testing Framework Tutorial (also linked from the tutorials section of eecs280.org). In brief, the tutorial will have you:

- Download `arrays.hpp` and `arrays.cpp` starter files with buggy code.
- Learn some new syntax and strategies for testing.
- Create tests in arrays_tests.cpp to catch bugs.
- Submit your tests to the autograder, which evaluates their thoroughness.

When you're finished, paste a copy of your `arrays_tests.cpp` on the next page.

Unit Testing Framework Tutorial (Continued)

Paste a copy of your `arrays_tests.cpp` into the box below. The tutorial has you submit to the autograder, but it's alright if you didn't necessarily catch all the bugs there. You'll get credit for completion on the lab as long as you've got some tests. Use tiny font so it fits :D

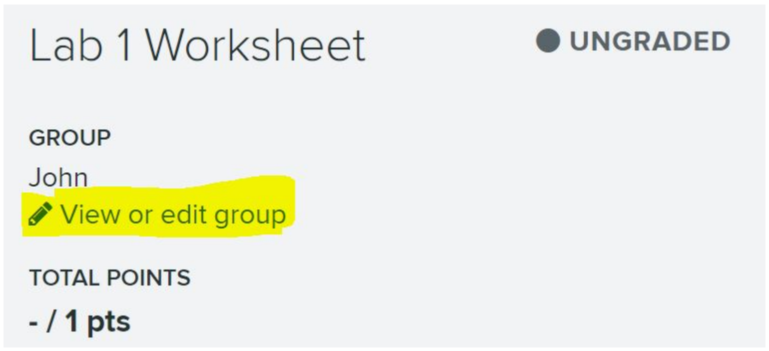
```cpp
#include "arrays.hpp"
#include "unit_test_framework.hpp"

// We define a test case with the TEST(<test_name>) macro.
// <test_name> can be any valid C++ function name.
TEST(test_slide_right) {
    int a[] = {1, 2, 3, 4, 5, 6};
    int b[] = {6, 1, 2, 3, 4, 5};
    slideRight(a, 6);
    for (int i = 0; i < 6; i++) {
        ASSERT_EQUAL(a[i], b[i]);
    }
}

TEST(test_flip) {
    int a[] = {1, 2, 3, 4, 5, 6};
    int b[] = {6, 5, 4, 3, 2, 1};
    flip(a, 6);
    for (int i = 0; i < 6; i++) {
        ASSERT_EQUAL(a[i], b[i]);
    }
}

TEST_MAIN() // No semicolon!
```

Submit Worksheet

Your pod is responsible for submitting lab worksheets to Gradescope by **before the deadline**. We strongly recommend submitting your worksheet on your lab day so you don't forget!

Download a PDF copy of your completed worksheet by going to *File > Download > PDF*.

The ***Scribe*** is responsible for submitting the PDF to Gradescope. **You need to add your pod members to the submission after it is uploaded:**



Lab 1 Worksheet

● UNGRADED

GROUP

John

View or edit group

TOTAL POINTS

- / 1 pts

Appendix: `p1_library.hpp`

```
#ifndef P1_LIBRARY_HPP
#define P1_LIBRARY_HPP

/* p1_library.hpp
 *
 * Libraries needed for EECS 280 project 1
 * Project UID 5366c7e2b77742d5b2142097e51561a5
 *
 * by Andrew DeOrio <awdeorio@umich.edu>
 * 2015-04-28
 */

#include <vector>
#include <string>

//MODIFIES: v
//EFFECTS: sorts v
void sort(std::vector<double> &v);

//EFFECTS: extracts one column of data from a tab separated values file (.tsv)
//  Prints errors to stdout and exits with non-zero status on errors
std::vector<double> extract_column(std::string filename, std::string
column_name);

#endif
```