Lab 1: Introduction to Lab Pods, Review

Due **Sunday Sep 10th, 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!

Getting to Know Each Other

In A → B → C → D order, say your name, favorite food, and why you're taking EECS 280! Write your responses below.

	Name	**Favorite Food**	**Why taking EECS 280**
Person A	Abhinav	Enchiladas	For the CS major
Person B	Zichen	Fries	For the DS major
Person C	Gage	Enchiladas	For robotics major
Person D	Michael	Dumplings	For CS major

Decide on a pod name. The ***Leader*** should lead this conversation.

Our pod name is...	Enchiladas & Fries & Dumplings

In C → D → A → B order, talk about one thing you're looking forward to in EECS 280 and the biggest challenge you expect this semester. Write your answer below.

	Thing you're looking forward to	**Biggest challenge you expect**
Person C	Learning more C++	Learning to play eurcre
Person D	Passing the class	The exams
Person A	Getting more C++ knowledge	Get an A in this class
Person B	Learn Linux commands	Achieve "A"s in both HISTORY 210 and HISTORY 241

How Lab Pods Work

Each pod member is assigned a different role for each meeting:

Leader	● Starts conversation when nobody knows what to talk about ● Makes sure the pod stays on task ● Encourages pod members to contribute equally (e.g. "[name], what do you think about this one?")
Scribe	● Makes an editable copy of the worksheet and shares it with the pod ● Writes the group's answers in the worksheet ● ***Responsible for submitting to Gradescope (+ adding pod members to the submission)*** ● Notes down questions to ask the staff member ● If the pod's answer to a question seems too simple, pushes the group to think deeper and explain more
Coder I Coder II	● Some lab exercises involve writing and running code outside of the worksheet. For these, the coder should share their screen. ● During code writing exercises, the coder should not do most of the work. Instead, their job is to get input from everyone on what to write. ● Coder I and Coder II alternate for different problems.

Labs are worth 10% of your course grade. Each lab group meeting is graded for **attendance** and **completion**. Your group submission to Gradescope is graded for completion and all group members earn the same score. However, individual scores are contingent on attendance, and students are unable to earn credit unless they attend the lab.

As a pod, talk over these scenarios to decide who is most responsible for helping the pod avoid these problems. The ***Leader*** should lead the discussion.

Problem	**Role that's responsible to prevent the problem**
One member dominates the discussion and others don't talk.	Leader
The worksheet is submitted with blank answers to the coding problems.	Coder & Scribe
The worksheet doesn't get submitted to Gradescope.	Scribe

Compile Time and Run Time Perspectives

Next, work through some exercises similar to what you can expect when we start more content-driven lab worksheets next week.

In lecture, we talked about the difference between the **compile time/source code** and **run time** perspectives. As a pod, write the definitions of these perspectives by having one person write a rough draft and the rest of the pod helping with refining the definition.

	Rough Draft Leader	**Definition**
Compile Time Perspective	Person C	It's the time during which the code is being compiled into machine code.
Run Time Perspective	Person D	The Runtime perspective displays the workflow execution details in graphical and tabular formats.

Many concepts in EECS 280 have different names depending on whether we are looking at them from the compile time or run time perspectives. Two examples are **variables** and **objects**. As a pod, write definitions for **variable** and **object**. Your definitions should include whether the term belongs to the compile time or run time perspective, as well as why it belongs there.

	Rough Draft Leader	**Definition**
Variable	Person B	a value that can change, depending on conditions or on information passed to the program
Object	Person A	an abstract data type with the addition of polymorphism and inheritance

Coding Exercise: Bugs!

The ***scribe*** and ***leader*** should look away for a moment. Do not look at the code below until the coders have done their part.

Coder I *and* ***Coder II*** should EDIT the code below to add one bug each (2 bugs total). The bugs should be interesting, but also realistic - don't make them obscure or overly complicated.

Then, the ***scribe*** and ***leader*** work together to find the bugs. Use the highlighter to mark the bugs once you've identified them.

```
// Return the number of values in a vector
// that are equal to the given value.
int count_if(vector<double> vec, double val) {

  int count = 0;

  for(size_t i = 1; i < vec.size(); ++i) {
    if(vec[i] = val) {
      ++count;
    }
  }

  return count;
}
```

	Why is this a realistic bug an EECS 280 student might make?	**Could the compiler identify this bug? Or would you need test cases?**
Bug 1 Highlight	Easy to forget that when comparing values there are two equal signs	Yes, the compiler can identify the bug.
Bug 2 Highlight	They are easy to forget and they are small enough to neglect.	No, I need to test cases

Coding Exercise: Loops

Write the function based on the specifications. **Coder I** is responsible for typing the code, but everyone is expected to contribute and work together!

```
// REQUIRES: nothing
// MODIFIES: cout and cin
// EFFECTS:  This function asks the user to type in a number between 5
//   and 9 (both inclusive). If the user enters an invalid number
//   (e.g. 10, -1, etc.), print an error message and prompt the user
//   again. Once the user enters a valid number, print a square of *s
//   where the dimension is the valid number the user typed in.
```

Your code here

```
void print_big_square() {
   cout << “Type a number between 5 and 9” << endl;
   int num;
   cin >> num;
   while (true){
      if(num <= 9 && num >= 5){
         return num*num;
      }
      else{
         cout << “Type a number between 5 and 9 PLEASE” << endl;
         cin >> num;
      }
   }
}
```

EECS 280 Strategy Brainstorm

Here's a quick break away from C++ material! Projects and exams make up large portions of your grade. You'll want a strategy for doing well on these and maximizing your scores! Each group member should brainstorm one (different!) strategy for doing well.

Your lab staff will stop by to say hi to your group. Find out their strategies, too.

		Strategy
Projects	Person A	Start early. Use as many submits as possible. Go to office hours and get help early.
Projects	Person B	Consider multiple edge cases and write as many test cases as possible before submitting
Projects	Staff	Use test-driven development
Exams	Person C	Test yourself on syntax and use piaza
Exams	Person D	Start studying early and go to office hours and use any resources to get help.
Exams	Staff	Challenge and check your understanding with Lobster and debuggers

Getting Help in EECS 280

One of the keys to success in EECS 280 is learning where to go when you need to figure something out, whether this is searching online or getting help from a staff member.

Here are some available resources:

- Piazza
- Lab pod student members
- Lab pod staff member
- Office hours
- Searching online (search engine, stackoverflow.com, online C++ references, etc.)
- Generative AI (e.g. ChatGPT)
- Debugging Tools: Visual Debugger, Valgrind, Address Sanitizer
- Proffice hours
- Request forms on eecs280.org
- Email professor or course staff (for non-course content/project related questions)
- EECS 280 Discord

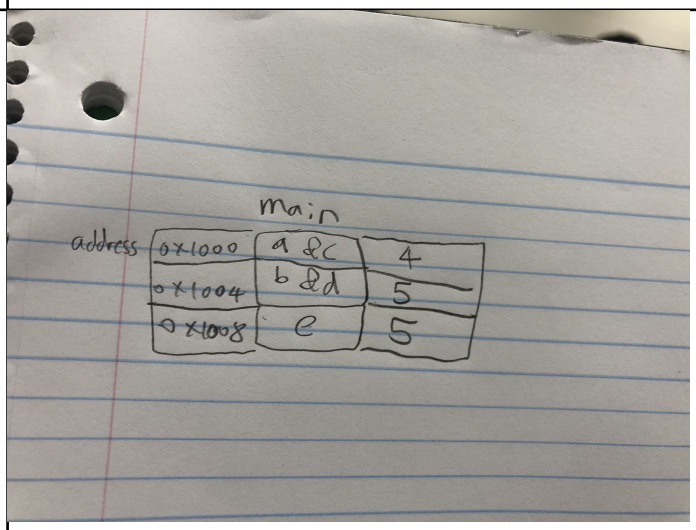
Consider the best resource(s) to use in the cases below. The ***Scribe*** records answers.

Question Type	**Best Resource to Use**
You get a compile error you don't recognize.	Piazza
You run into an error when setting up your computer for C++.	Office Hours
You're not sure where to start on a project.	Piazza
Your program's output doesn't match up on the public test.	Piazza
You're concerned about your grade in the course.	Proffice Hours
Your project crashes with a segmentation fault and you're not sure where.	Piazza/Office Hours
You have a medical or personal emergency.	Email professor or course staff
The project just seems way too complicated.	EECS 280 Discord
Your code passes the public tests in your IDE but fails them on the Autograder.	Office Hours

Value Semantics and References

Draw a memory diagram of the following program (which you will use to answer some questions below). ***Coder II*** 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`
`}`
	

Answer the following questions about the code and its behavior at runtime. The ***Leader*** should lead the discussion.

Question	**Answer**
How many variables are there?	5
How many objects will there be?	3
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;`)?	3
Is the diagram you drew from the source code or the run time perspective?	Run time perspective

Optional: Set Up Pod Group Chat

We encourage creating a GroupMe/WeChat group/Slack workspace/etc. for your pod as a space to communicate, ask questions, and collaborate!

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:**