Working in Groups

The primary purpose of the Discussion meetings in PIC16A is for you to work in small groups on programming problems. Working in groups allows you to:

- Learn programming skills from your classmates.
- Build your skills as a peer teacher.
- Help you practice verbal reasoning about programming solutions.

As a reminder, **you are required to attend discussion sections unless you have made an alternative arrangement approved by me.** Your attendance in discussion sections is measured by your completion of an in-class assignment, and is worth a major component of your final grade. The primary acceptable reasons for alternative arrangements are:

- The discussion section takes place either after 8pm or before 8am in your local time.
- You are not able to reliably access the internet during the scheduled discussion section.

In the event of timezone-related difficulties, I will attempt to match you with other students in similar situations. You will then be responsible for meeting with these students outside of the scheduled discussion section to complete the in-class assignment.

In this document, we'll discuss how we expect you to work on your in-class assignments in PIC16A.

Three Roles

A typical group will have three students. We will do our best for your group assignment to reflect your interests as indicated in the PIC16A Entrance Survey (https://docs.google.com/forms/d/e/1FAIpQLSfLc5H2wxtgQX8dXxuv1h_ij8kua3QE57USd9cbkM1hi_usp=sf_link) (fill it out!).

In each discussion section:

- One group member will be the **Driver.** The Driver takes suggestions from the Proposer and Reviewer and turns them into functioning code.
- One group member will be the **Proposer.** The Proposer suggests high-level approaches to problems, keeping in mind the long-term needs of the assignment.
- One group member will be the **Reviewer.** The Reviewer gives feedback on the Proposer's suggestions, and also gives suggestions on how the Driver can improve the code.

The Driver

The **Driver** is the primary writer of code during the discussion section. The Driver is the only team member who types code directly into a notebook. Because of this, the Driver should be sharing their screen with the other group members.

The Driver is responsible for taking the high-level, strategic suggestions of the Proposer and Reviewer and implementing functioning code. If the Proposer and Reviewer suggest "loop through the characters of `s` ", it is the Driver's responsibility to remember how to do this, and to write a line like

```
for c in s:
    ...
```

As the Driver, you can always ask the Proposer or Reviewer for suggestions about what to do next, and reminders if you have forgotten a piece of syntax. However, they cannot write the code for you -- all code in the assignment is written by you.

The Driver is also responsible for downloading the section worksheet and turning it in. Before turning it in, the Driver must write the names of the team members **and their roles** at the top of the assignment. If a team member is absent, the Driver should note this.

The Proposer

The **Proposer** is the initiator of ideas on the team. As Proposer, your primary responsibility is to come up with high-level suggestions for the Driver about what to do next. The Proposer should be thinking several steps ahead: "after we write the loop, we should then handle possible type errors and..."

The Proposer should always verbalize their suggestions, and pose them to the Reviewer for feedback prior to implementation by the Driver.

The Proposer should avoid thinking that they have "the" correct answer -- many problems can be solved by multiple methods.

The Reviewer

The Reviewer is responsible for offering **constructive** feedback on the suggestions of the Proposer and the implementations of the Driver. When the Proposer offers a suggestion, the Reviewer ask questions in order to make the Proposer's intention clear. The Reviewer should then aim to offer at least one positive and one negative aspect of the proposed approach. The Proposer and Reviewer should then evaluate whether to move forward, or whether to quickly think of some alternative approaches which can then be evaluated.

Like the Proposer, the Reviewer should be careful not to assume that they are "right" or the Proposer "wrong" when disagreement arises.

In the event that the Reviewer and Proposer cannot agree on a course of action, they should ask the Driver for input.

Ask for help

At any time, if you are experiencing difficulty in executing your role, ask a teammate for help! Can't remember a particular piece of Python syntax? Not sure what kind of data structure is best for the task? Can't find a potential negative aspect of a suggestion? Ask your group members to give you a hand.

Patterns and Anti-Patterns

In this section, we'll go over a few patterns: good practices which you should emulate while working together. We'll also show a few anti-patterns for you to avoid.

Zenith, Victor, and Eun are working together in a group. This time, Xenith is the Driver, Victor is the Proposer, and Eun is the Reviewer.

Patterns (emulate)

Example 1

The normal flow of work should sound something like this.

- **Victor** (Proposer): I suggest that we use a dictionary to store pairs of names and favorite ice-cream flavors.
- **Eun** (Reviewer): So in this dictionary, the keys would be names?
- **Victor**: Yes, that's my idea.
- **Eun**: I like this idea because dictionaries are perfectly suited for looking up related information, but don't we need to look up the list of all students who prefer chocolate ice cream later in the assignment?
- **Victor**: Great point Eun. We could have each flavor be a key, with value giving the list of all students who prefer that ice cream.
- **Eun**: That sounds good to me! Xenith?
- **Zenith** (Driver): Yeah, that makes sense to me too. Here, I'll code it up. Eun, tell me if you spot any issues!

Example 2

Get stuck? Ask for help!

- **Victor** (Proposer): Hey, so I know we need to keep track of the number of times this function has been called, but I can't remember how. Can either of you help me out?
- **Eun** (Reviewer): Oh, yeah...we discussed this, but I'm drawing a blank. Zenith?
- **Zenith** (Driver): I think we can do this with either global variables or defining a custom class.
- **Victor**: Thanks Zenith! Eun, do you agree?
- **Eun**: Yes, custom classes require more code to write but are much safer.
- **Zenith**: Great. I'll call it `my_counter` and give it a `count_calls()` method like this...

Anti-patterns (avoid)

Too directive

Don't veto your partner's suggestions.

- **Victor** (Proposer): Could we use a dictionary here?
- **Eun** (Reviewer): No, let's use two lists instead.

- **Eun** (Reviewer): I think the `if` statement would work, but could we get more informative error messages with a `try` statement?
- **Victor** (Proposer): No, let's stick with `if`.

If you think you have a better idea, state **why** you think it might be preferable, and see if you can convince your partner.

It's especially important that the Driver not ignore their partner's suggestions.

- **Victor** (Proposer): What if we added an additional case for when `x = 0.0`.
- **Zenith** (Driver): No, let's move on.

Disparaging your partners or their suggestions

Don't do it.

- **Eun** (Reviewer): Instead of a list, wouldn't it be simpler to use a set?
- **Victor** (Proposer): That's dumb, sets can't store repeated values.

Checking out

It's important that all partners be actively contributing to conversation throughout your meeting. Stay engaged. You are **encouraged** (but not required) to turn your cameras on in order to help connect with each other.

LAs

The LAs will be rotating through groups in each discussion section, in order to help foster strong team communication and ensure that that Driver, Proposer, and Reviewer are each actively engaged and effectively carrying out their responsibilities.

Rotating Roles

The first session in which we will use this format is Thursday, October 8th (on Tuesday 10/6 we will practice). Within your group, sort yourselves alphabetically by first name. For example, Eun, Victor, Zenith. On Tuesday October 8th, the first group member (Eun) is the Driver, followed by Proposer Victor and Reviewer Zenith. Next time, the Driver becomes the Proposer, the Proposer becomes the Reviewer, and the Reviewer becomes the Driver. So, on Tuesday, October 13th, Victor would be the Driver, Zenith the Proposer, and Eun the Reviewer.

On Zoom

When working in groups in your discussion section, it is important to engage with your group members as peers, mentors, and individuals. You are **strongly encouraged** to have your cameras on when working with your teams. Please put some effort into making this possible. LAs rotating through the groups may ask you why your camera is off.

Contingencies

Timezones

If the discussion section falls outside of the hours 8am-8pm in your local time, I will attempt to match you with a group of students in similar circumstances. The three of you will be responsible for arranging a meeting of no less than 30 minutes to work on the assignment outside of class time, on the day that the discussion section meets.

Missing group member

If one of your group members is missing, please note this on your assignment. The two remaining group members should work as a pair, with one playing the role of Driver and one playing the role of Proposer. Both should offer feedback and suggestions for improvements (i.e. they each play the role of the Reviewer as well).

In [ ]: