**How to get the starting code for a Pair Programming project**

1. When you arrive in class, find a partner with whom you would like to pair program. Your instructor will help you as needed.  
   ***Do NOT continue until your instructor tells you to do so.***
2. When your instructor tells you to proceed (and NOT BEFORE THEN), your pair (or triple) should ***select a name*** that you think will be unique to your team, ***that begins with your section number,*** like this example for a team in Section 2:

**2-bob-jill**

Use that name in the following.

1. Decide which of the pair will be Person 1 and which Person 2 (and which will be Person 3 if working in a team of three).
2. **Person 1** (and ONLY Person 1):
   1. Follow the link to the Starting Code for Pair Programming for the session, for example this [link for Session 3](https://classroom.github.com/a/oi1c9Cmv).
   2. At the link you will see something like this:

Graphical user interface, text, application, email

Description automatically generated  
**Person 1 (and ONLY Person 1) should then *type your UNIQUE team name* and then press the *Create Team* button, as shown above.**

* 1. Person 1 will then see the usual “Accept the [group] assignment…” message and should continue in the usual way to ***clone the project into PyCharm***.

1. Now **Person 2** (and **Person 3** if applicable) should:
   1. Follow the link to the Starting Code for Pair Programming for the session, for example this [link for Session 3](https://classroom.github.com/a/oi1c9Cmv).
   2. At the link you will see something like this:



Person 2 (and Person 3 if applicable) should ***find their team name*** and then press its ***Join*** button, as indicated above. **BE CAREFUL** – select YOUR team name. THERE IS NO “UNDO” option!

* 1. Person 2 (and Person 3 if applicable) will then see the usual “Accept this [group] assignment…” message and should continue in the usual way to ***clone the project into PyCharm***.

1. The group then ***pair-programs***, using ONE computer at any instant in time. Each time you switch Driver/Navigator, the old Driver does a **Git ~ Commit-and-push** to send the current version to the repository and the new Navigator does a **Git ~ Pull** to obtain the updated version of the project.
2. When done, commit-and-push and/or pull your work. Then each team member ***manually copies the work from their shared repository to their individual repository*** for that session, and turns in their individual repository as usual (by commit-and-push and also via the Moodle quiz/dropbox).