2017-09-05 Lab

Introductions

- What is your name?
- What year are you?
- What is your major/minor?
- Why are you taking this course? It's okay to say that you just wanted to fulfill a requirement.

Community Standards

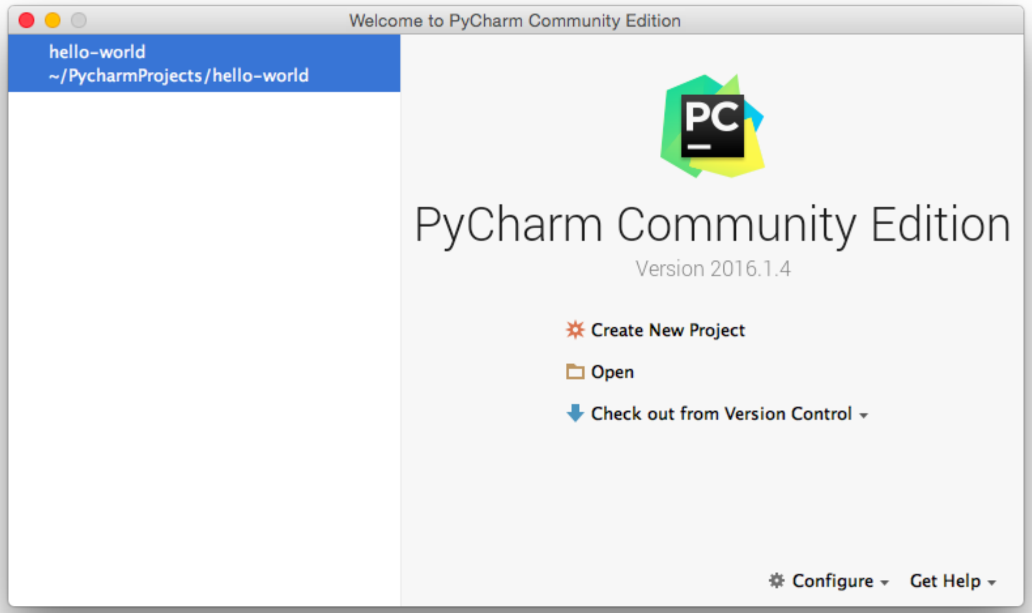
- What are some guidelines for working with each other?
- How will you resolve disputes?

We will have a discussion about this, but please email your guidelines to me (justinnhli@oxy.edu). I will compile them and have you sign a copy next week.

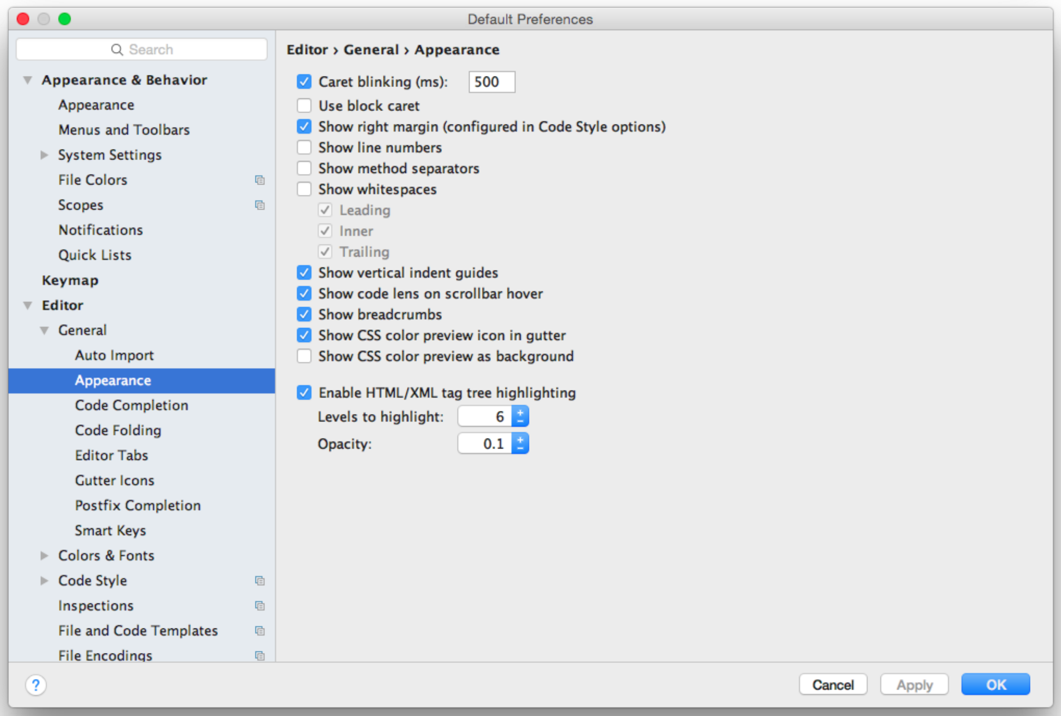
Software (Python and PyCharm)

1. Download and install Python 3 from the Python website.
2. Download and install PyCharm Community Edition from the PyCharm website.

PyCharm has a lot of settings that you can play with, but we will only set one for now. When you first start PyCharm, you should see a screen like this:


Welcome to PyCharm Community Edition
hello-world
~/PycharmProjects/hello-world
PyCharm Community Edition
Version 2016.1.4
Create New Project
Open
Check out from Version Control
Configure
Get Help

Click on "Configure" on the bottom right, then "Preferences". On the left, go to "Editor", then "General" and "Appearance". You should see a window like this:



Editor › General › Appearance

- [✓] Caret blinking (ms): 500
- [ ] Use block caret
- [✓] Show right margin (configured in Code Style options)
- [ ] Show line numbers
- [ ] Show method separators
- [ ] Show whitespaces
  - [✓] Leading
  - [✓] Inner
  - [✓] Trailing
- [✓] Show vertical indent guides
- [✓] Show code lens on scrollbar hover
- [✓] Show breadcrumbs
- [✓] Show CSS color preview icon in gutter
- [ ] Show CSS color preview as background
- [✓] Enable HTML/XML tag tree highlighting
  - Levels to highlight: 6
  - Opacity: 0.1

Make sure that "Show Line Numbers" is checked.

Your First Program

1. Create a new project in PyCharm. You can choose where you want your projects to be saved, but please remember this location, as you will need it in the future. In general, you should start a new project for every lab, homework, and project. If you write code as part of your lecture notes, I also encourage you to start a new project for every lecture. This keeps everything tidy and makes it harder to accidentally run the wrong program.

2. If you are asked for the Interpreter, select the one that says "Python 3".

3. In your new project, create a file called `hello.py`. You can do this using the "File" menu, then "New", then "Python File".

4. In `hello.py`, type

   ```
   print("Hello, World!")
   ```

   then save the file.

5. From the menu, select Run > Run. If asked what to run, select "hello.py".

6. You should see a new panel on the bottom that says "Hello World!".

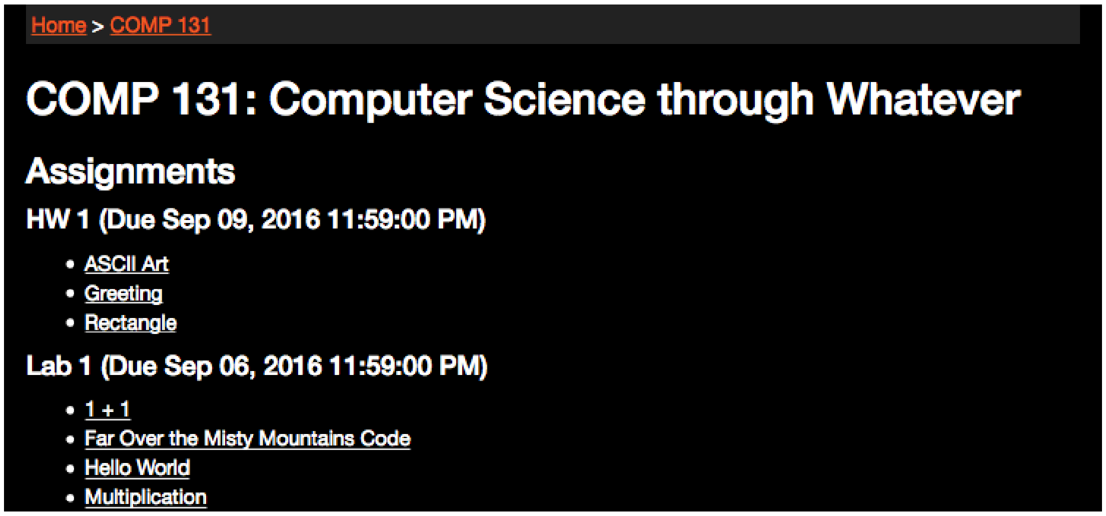
Congratulations! You just wrote your first computer program!

The Autograder

1. Open your browser, and go to the autograder at autograder.cc.oxy.edu. Log in using your Oxy account.

2. Wait for me to catch up. I have to manually enroll you into this semester's COMP 131 course.

3. Refresh the page, and look for "COMP 131" on the sidebar. You should see something like this:



Home > COMP 131

COMP 131: Computer Science through Whatever

Assignments

HW 1 (Due Sep 09, 2016 11:59:00 PM)

- ASCII Art
- Greeting
- Rectangle

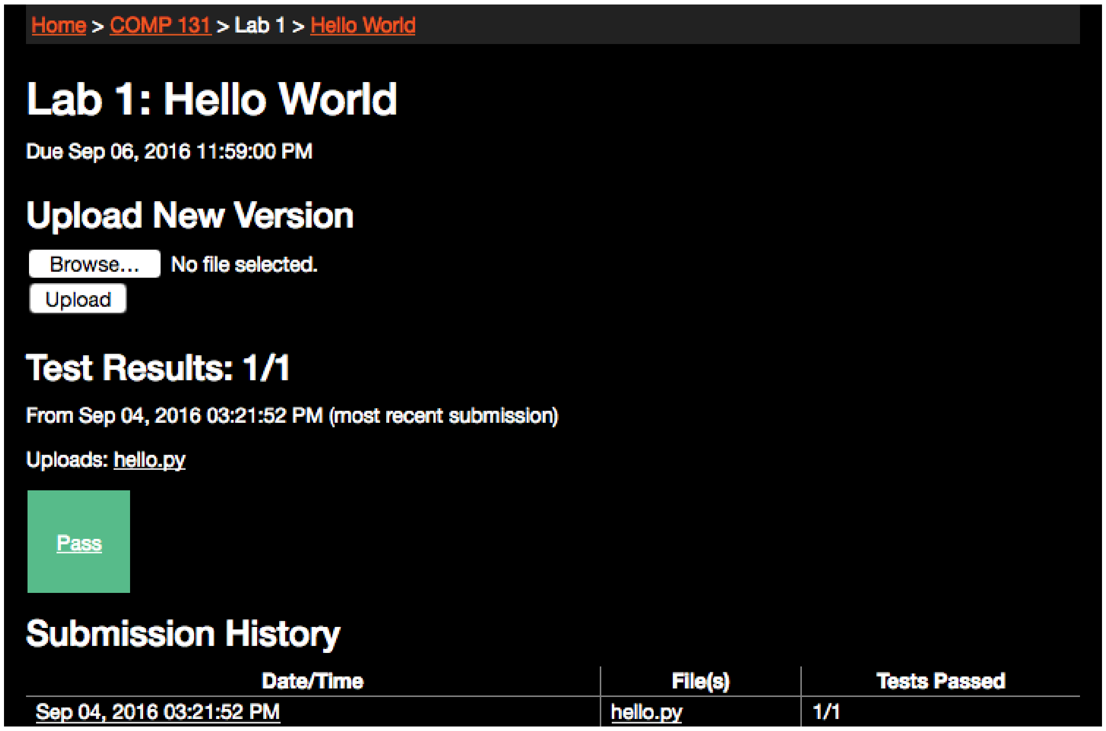
Lab 1 (Due Sep 06, 2016 11:59:00 PM)

- 1 + 1
- Far Over the Misty Mountains Code
- Hello World
- Multiplication

4. Click on the "Hello World" project - this is where you will submit your Hello World program so I can check whether your code works.

5. Click on "Browse", then find the `hello.py` file that you created. Submit the file.

6. Refresh the page. You should see something like this:



Home > COMP 131 > Lab 1 > Hello World

Lab 1: Hello World

Due Sep 06, 2016 11:59:00 PM

Upload New Version

Test Results: 1/1

From Sep 04, 2016 03:21:52 PM (most recent submission)

Uploads: hello.py

Submission History

Date/Time	File(s)	Tests Passed
Sep 04, 2016 03:21:52 PM	hello.py	1/1

Notice the colored box in the middle - these represent individual *test cases*, which is how I make sure your program is correct. This "Hello World" project only has a single test case, but later projects will have tens, if not hundreds of test cases.

7. Click on the test case - you should see what the program is expected to print, and what your program actually printed. These must match for the test case to pass. Be careful! The autograder is nitpicky - you have to get every letter in the correct case and every period and comma in the right place to pass a test case. It is a little less strict about spaces at the beginning and end of lines.

8. Go back to `hello.py` in PyCharm. Change the program to print something else, then submit it again. You should see the test case fail. Play around with the autograder to make sure you are comfortable with how it works.

9. Note that you can only submit to the autograder once every five minutes (for each project). This time limit exists to encourage you to double-check your own code before submitting, instead of relying on the autograder to tell you whether you are correct.

10. When you are done, submit a correct version of your code. Unless stated otherwise, you are graded by your *last* submission on all labs, homeworks, and projects. It is your responsibility to make sure that the correct file is submitted, and that it is submitted before the deadline.

More Practice

Complete the following programs in a new file and submit them to the autograder.

1. Write a program that prints out the result of `1 + 1`. DO NOT just write `print(2)`.

2. Write a program that prints out the result of `1 * 3 * 5 * ... * 17 * 19`.

3. Write a program that prints the following:

   ```
   Far over the Misty Mountains cold
   To dungeons deep and caverns old.
   We must away ere break of day,
   To claim our long-forgotten gold.
   ```

A Note on Coding Style

As I mentioned in class, code is mostly written for other people to read. Here are some hints on how to make your code more readable. Keep this in mind as the semester progresses.

- Put you and your partner's name in a comment at the top of the file. Feel free to put additional information as well (eg. date, project, etc.)

- Comment your code. Anything that could be unclear to the reader, or to future-you in two weeks, deserves an explanation in comments. Don't over do it though - you don't need a comment that says `# adds 1 to x`. One exception to this is if the formula is well known. For example:

  ```
  # convert Fahrenheit to Celsius
  celsius = 5.0 / 9.0 * (fahr - 32.0)
  ```

- Stick to one variable naming convention (`under_scores` or `camelCase`)

- Put constants into variables with meaningful names in `UPPER_CASE`. The result of not following this rule are "magic numbers" - numbers that appear in code without explanation.

- Use parenthesis where the order of operations may be unclear. Put a space around operators. Instead of this:

  ```
  count=(a+b)*c/d+random()
  ```

  do this:

  ```
  count = (a + b) * c / d + random()
  ```

- In general, apply your own judgment as to what is understandable and what isn't. If the line is to long, break it down into meaningful parts and assign them to variables.
- Avoid repeating code. If you noticing yourself copying and pasting code, you should probably write a function instead. (You'll learn how to do this next week.)

Documentation

One reason we are learning Python in this class is that there is a lot of documentation online about how to write Python. Learning to read this documentation is essential, since new programming languages come and go, and even in the same language you will often use code that you didn't write.

1. Go to the section of the Python documentation for "binary arithmetic operators".
2. What does the `%` operator do? That is, if I write `print(5 % 2)`, what is printed? What if I change the numbers?
3. What does the `//` operator do? That is, if I write `print(5 // 2)`, what is printed? What if I change the numbers?
4. How do you exponents (eg. 3^2, 2^{10}) in Python? (Hint: google this!)

Although the explanations make seen arcane at first, with a little practice you will have no problem understanding what they are saying.

Once you have told me (in person) the answers to the above questions, you are done with lab for the day.