OIM3640 – Problem Solving and Software Design

2022 Spring

Session 06 (2/03)

Today's Agenda

1. Welcome/News/Announcements

2. Class review

3. Revisit exercises

4. More practice

5. Lecture

   ○ Functions II – Case: Interface Design

Welcome/News/Announcements

1. Exercises
   - 35/43 **finished and pushed** to GitHub on time
2. **No** IN-PERSON/ONLINE **class** on 2/15
3. Office Hours
4. Questions?

What we have learned so far...

- Variables, Expressions, Statements

- Types: int, float, string, boolean, `None`, other data structures

- Functions:
  - syntax:

    ```python
    def f():
        """ docstring """
        # function body
    ```

  - Variables and Parameters are **local**

  - Discussions on `print()` vs. `return` : 1, 2, 3, ...

Quick Quiz

Answer the following question(s):

1. Can function return **multiple** values?

2. How do we check if variable `a` is an integer?
   - `isinstance(a, int)`

3. When we don't know what to do to finish the function, what shall we use?
   - comments!
   - `pass`

4. What will be the output below?

```python
def f():
    print('Hi')

print(f())
```

Exercise Feedback

- Please check your OIM3640/**Issues** on GitHub.
  - **Reply** if we need to continue the conversation.
  - **Comment (with evidence) and Close** if you think it is fixed.
- File/folder names - use **lowercase** and **underscore**, i.e., *session04/type_demo.py*
- Use more Python **comments** and be ready to write **pseudo-code** and **docstrings**
- If you have any question regarding
  - Class **demo code**, create new issue in OIM3640/oim3640
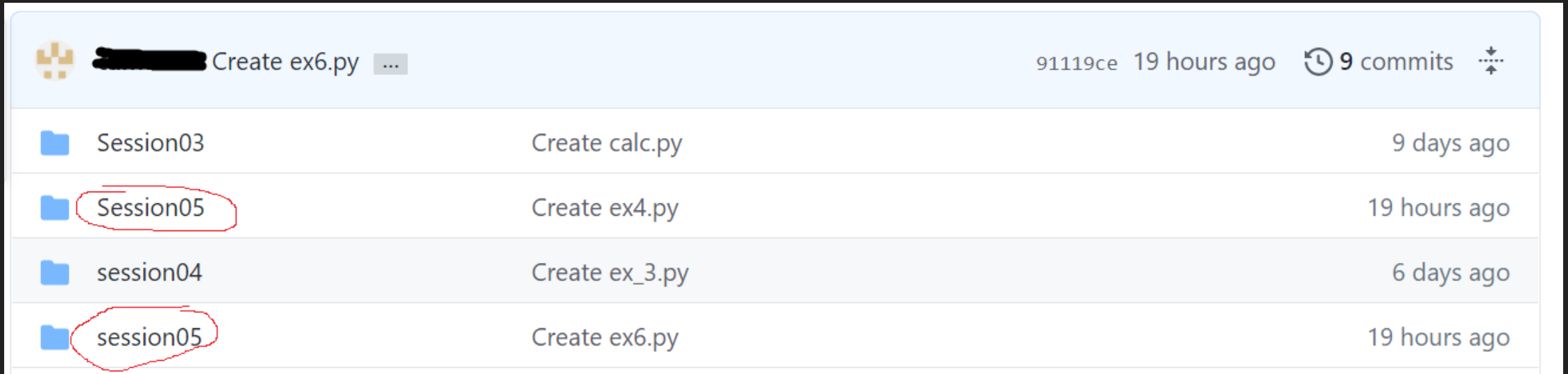 - Class **material**, create new issue in OIM3640/resources

Docstring! Docstring! Docstring!

How to fix this

1. Move the folder out of repo.

2. Commit

3. Move the folder back.

What if exercises are too difficult?

1. Do I understand the concepts?

2. Have I attempted example/demo code?

3. Have I read supplemental materials?

4. Have I tried any debugging approaches?

5. Have I had enough rest?

6. Have I asked for help?

    i. internal community (slack/GitHub/email)

    ii. external community

        ■ Please give credits for use of code.

        ■ Cite URLs as references in comments.

Revisit Exercises

- Exercise 3-5:
  - `my_abs()` : create your own version of built-in `abs()` , i.e. without using `abs()`
  - Argument checking - stackoverflow discussion
  - How does **module** work?
  - Use **Psuedo-code**
- Exercise 6:
  - `quadratic()`

Revisit Exercises

-
```python
def quadratic(a, b, c):
    a = float(input('Enter a: '))
    b = float(input('Enter b: '))
    c = float(input('Enter c: '))
    ...
```

- `return(-x)` -> `return -x`

- `if some_condition == True:` -> `if some_condition:`

- Make sure you test (aka. **call**) the function.

Python Tips

- Always format your code before running.
- Use Python Tutor
  - to visualize code execution
  - Please bookmark this website: https://pythontutor.com/visualize.html

Practice

1. **Codingbat** (OIM3640/codingbat)
2. Python Challange
    i. Challenge 1
3. More learning resources

Lecture

- Functions II - Case: Interface Design
  - **Watch Videos** together
  - **Read** class materials, and **try all the examples** in your VS Code
  - **Attempt** exercises together
  - **Complete** exercises **before due time**