
Fall 2021 - Section 002 - In-person - 11am

Class 05 - Control Flow Practice

# Today's Goals

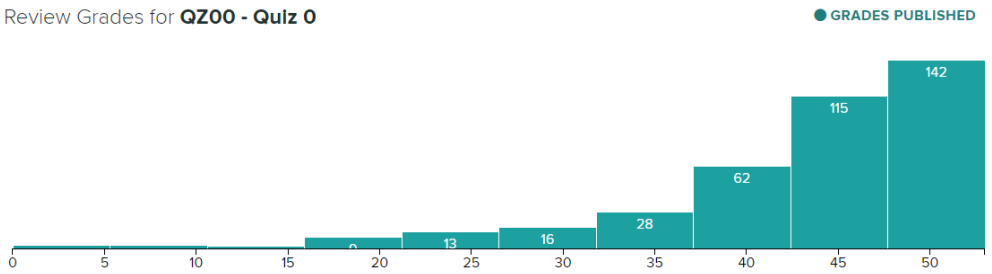
1. Announcements

2. Practice and review control flow statements (if-then and while)

3. Learn bool Operators: and, or, not

# Quiz 0 - Scores Released

- Quiz 0 Scores Released: 85 median, 82 mean
  - Compare with Fall Quiz 0 Scores: 84 median, 81 mean




- **Diagram Penalties - Points returned for Quiz 00 Only - Upload PDFs correctly!**
- **Regrade Requests** ARE NOT for asking WHY something was marked wrong -- open through 9/8 at 11:59pm
  - If you have a question about why something was marked wrong: office hours or tutoring!
  - Regrade Requests are open through Tomorrow at 11:59pm EST. The time to understand what went wrong is now, not later. We will not accept late regrade requests.
  - Request a Regrade on the specific question marked wrong.
  - For instructions, read: **http://bit.ly/regrade-request**

# Quiz 0 - Reflections

- Pay close attention to the details of concepts
  - General number of concepts in 110 is small, but their details are important
  - Skimming readings or 2x'ing videos without engaging will bite you
    - If you skimmed LS05 Objects and Type and LS06 Expressions... go back and read them!
  - We expect close to 100% on Lesson assignments, but these are a low bar
  - You are encouraged to self-study by tinkering and exploring beyond lecture

- Tonight Week-in-Review with Kaki @ 4pm EST in SN014
  - Will review quiz
  - Zoom link will post on Itinerary Tomorrow
  - Recording will post after

- Come work with us in tutoring, or office hours if you can't make it!

# Tutoring continues tonight!

- **Tutoring is your place for conceptual help at a personalized pace**, there is no time limit to interactions like Office Hours
- **Tuesday through Thursday 5-7pm in FB007**; full schedule and Zoom link located at course.care
- You can have **1-on-1 interactions with TAs or visit with a group** of other students to go over similar concepts
- Great place to **review lecture material and in-class exercises, go over quizzes, or study** for upcoming quizzes
- **No exercise help** is offered through tutoring, visit Office Hours for all Exercise and Project questions

# Other Announcements

- Undergraduate Teaching Assistant (UTA) E-mail Contacts Assigned!
  - Sakai > PostEm (under announcements) > Email Pairings
  - They'll be introducing themselves within the week

- E-mail Best Practices
  - If you need to describe your code, some error you are seeing, or have an impulse to screenshot something going wrong... come to office hours!
  - If you have a question about course logistics, a one-off non-emergency absence, or a specific conceptual question, e-mail your assigned UTA!
    - If they have not responded after 24 hours, forward to comp110-heads@googlegroups.com
  - If you have a hospitalization, death in the family, or comparable emergency
    - Do what you need to do for you and your family first and foremost, then email comp110-heads@googlegroups.com

# Warm-up Challenge Questions #1

Warning: this code listing contains logical errors.

```python
"""Challenge Question #1"""

choice: int = int(input("Enter a number: "))

if choice > 50:
    if choice < 25:
        print("A")
    else:
        print("B")
else:
    if choice > 75:
        print("C")
    else:
        print("D")
```

```python
"""Challenge Question #1"""

choice: int = int(input("Enter a number: "))

if choice > 50:
    if choice < 25:
        print("A")
    else:
        print("B")
else:
    if choice > 75:
        print("C")
    else:
        print("D")
```

# Warm-up Challenge Questions #2

```
"""Challenge Question #2"""

i: int = 0
s: str = ""

while i < 4:
    if i % 2 == 0:
        s = s + "h"
    else:
        s = s + "e"
    i = i + 1

print(s)
```

```python
"""Challenge Question #2"""

i: int = 0
s: str = ""

while i < 4:
    if i % 2 == 0:
        s = s + "h"
    else:
        s = s + "e"
    i = i + 1

print(s)
```

# Boolean Operators

not bool_r

Invert/flip T to False and vice-versa.

| | result |
|---|---|
| T | F |
| F | T |

bool expression

b_lhs and b_rhs

True iff both bools on lhs and rhs evaluate to True.

| lhs \ rhs | T | F |
|---|---|---|
| T | T | F |
| F | F | F |

b_lhs or b_rhs

True iff either lhs or rhs is True or both are True

| lhs \ rhs | T | F |
|---|---|---|
| T | T | T |
| F | T | F |

# Precedence

high

( )

Arithmetic { ** ; * and / ; + and - }

Relational

not

and

or

low

good practice to use parens!

True or (True and (not A))