

Testing

- Video: Testing Automatically Using doctest 6 min
- Reading: Testing Automatically Using doctest 10 min
- Video: Writing a '__main__' program 4 min
- Reading: Writing a '__main__' program 10 min
- Video: Creating Your Own Types 6 min
- Reading: Creating Your Own Types 10 min
- Video: Testing Automatically Using unittest 4 min
- Reading: Testing Automatically Using unittest 10 min
- Video: Choosing Test Cases 7 min
- Reading: Choosing Test Cases 10 min
- Video: Testing Functions that Mutate Values 3 min
- Reading: Testing Functions that Mutate Values 10 min

Review

- Quiz: Week 2 Exercise 10 questions

Assignment 1

- Peer-graded Assignment: Writing test cases 1h
- Review Your Peers: Writing test cases

QUIZ • 30 MIN

Week 2 Exercise

Submit your assignment

DUE DATE Aug 23, 2:59 PM CST ATTEMPTS 3 every 8 hours

Receive grade

TO PASS 80% or higher

Week 2 Exercise

TOTAL POINTS 10

1. Consider this code:

```
1 def high_low(guess, actual):
2     """ (int, int) -> str
3
4     Return "your guess is low" if guess is lower than actual,
5     "your guess is High" if guess is higher than actual, and
6     "correct" if guess is equal to actual.
7
8     >>> high_low(4, 10)
9 Grade "your guess is low"
10 _
```

1 point

Which set of test cases is the *best choice* of tests cases for this function? (Think about the **Boundaries** category for choosing test cases.)

- ☐ • **guess** refers to "**higher**" and **actual** refers to "**lower**"
- ☐ • **guess** refers to "**correct**" and **actual** refers to "**correct**"
- ☐ • **guess** refers to "**lower**" and **actual** refers to "**higher**"
- ☐ • **guess** and **actual** refer to negative values
- ☐ • **guess** and **actual** refer to zero
- ☐ • **guess** and **actual** refer to positive values
- ☒ • **guess** refers to a value that is less than the value referred to by **actual**
- ☐ • **guess** refers to a value equal to the value referred to by **actual**
- ☐ • **guess** refers to a value that is greater than the value referred to by **actual**
- ☐ • **guess** and **actual** refer to odd values
- ☐ • **guess** and **actual** refer to zero
- ☐ • **guess** and **actual** refer to non-zero even values

2. Consider this code:

```
1 def first_two_items(L):
2     """ (list of str) -> list of str
3
4     Return the first two items in L. If there are fewer than
5     two items in L, return all of the items.
6
7     >>> first_two_items(["apple", "pear", "grape"])
8     ["apple", "pear"]
9     """
```

1 point

Which of these sets of values for **L** is the *best choice* of tests cases for this function? (Think about the **Size** category for choosing test cases.)

- ☒ • []
- ☐ • ["one"]
- ☐ • ["one", "two"]
- ☐ • ["one", "two", "three", "four"]
- ☐ • []
- ☐ • ["one"]
- ☐ • ["one", "two"]
- ☐ • ["one", "two", "three"]
- ☐ • ["one", "two", "three", "four"]
- ☐ • []
- ☐ • ["one"]
- ☐ • ["one", "two"]
- ☐ • ["one", "two", "three"]
- ☐ • ["one", "two", "three", "four"]
- ☐ • ["one", "two", "three", "four", "five"]
- ☐ • ["one", "two", "three", "four", "five", "six"]
- ☐ • More tests like this, with lists up to 20 items long.

3. Consider this code:

```
1 def is_preschooler(age):
2     """ (int) -> bool
3
4     Precondition: age >= 0
5
6     Return True if and only if age is between 3 and 5
7     inclusive.
8
9     >>> is_preschooler(4)
10    True
11    """
```

1 point

Which set of test cases is the *best* choice of tests cases for this function? (Think about which test case category you might want to consider.)

- ☒ • 0 or 1
- ☐ • 2
- ☐ • 3
- ☐ • 4
- ☐ • 5
- ☐ • 6
- ☐ • one number over 6
- ☐ • one number under 0
- ☐ • 0
- ☐ • one number over 0
- ☐ • one number between 0 and 2
- ☐ • 4
- ☐ • one number over 5
- ☐ • one number between 0 and 2
- ☐ • one number between 3 and 5
- ☐ • one number over 5

4. Consider this code:

```
1 def count_occurrences(s, ch):
2     """ (str, str) -> int
3
4     Precondition: len(ch) == 1
5
6     Return the number of occurrences of ch in s.
7
8     >>> count_occurrences("hello", "l")
9     2
10    """
```

1 point

Which of these sets of values for **s** and **ch** is the *best choice* of tests cases for this function? (There are several test case categories for this question, including at least **Size** and **Dichotomies**.)

- ☒ • **s** refers to "", **ch** refers to 'a'
- ☐ • **s** refers to 'a', **ch** refers to 'a'
- ☐ • **s** refers to 'a', **ch** refers to 'b'
- ☐ • **s** refers to 'abc', **ch** refers to 'b'
- ☐ • **s** refers to 'abc', **ch** refers to 'd'
- ☐ • **s** refers to 'abcabca', **ch** refers to 'a'
- ☐ • **s** refers to "", **ch** refers to 'b'
- ☐ • **s** refers to 'a', **ch** refers to 'b'
- ☐ • **s** refers to 'aaaaaa', **ch** refers to 'b'
- ☐ • **s** refers to '', **ch** refers to 'a'
- ☐ • **s** refers to 'a', **ch** refers to 'a'
- ☐ • **s** refers to '1', **ch** refers to '1'
- ☐ • **s** refers to 'a', **ch** refers to 'b'
- ☐ • **s** refers to 'abc', **ch** refers to 'b'
- ☐ • **s** refers to '123', **ch** refers to '2'
- ☐ • **s** refers to 'abc', **ch** refers to 'd'
- ☐ • **s** refers to 'abcabca', **ch** refers to 'a'
- ☐ • **s** refers to "", **ch** refers to 'a'
- ☐ • **s** refers to 'a', **ch** refers to 'a'
- ☐ • **s** refers to 'aaaaaa', **ch** refers to 'a'

5. Consider this code:

```
1 def can_vote(age):
2     """ (int) -> bool
3
4     Precondition: age >= 0
5
6     Return True if and only if a person aged age can vote
7     in Canada. The legal voting age in Canada is 18 years and
8     older.
9     """
10
11    return age > 18
```

1 point