User Manual

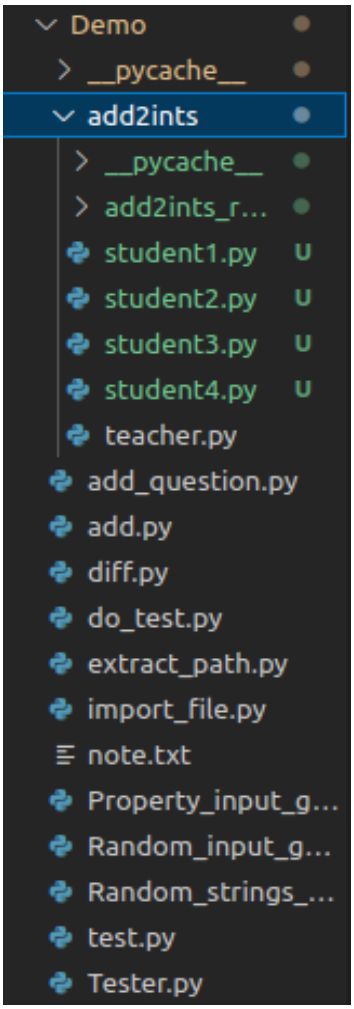
CMPT416-Spring 2023

Qingrui Li -301400099

April 11, 2023

Position of the demo: https://github.com/manyasharma01/CMPT-416/tree/main/Rachel_416/Demo

Meaning of each file in Demo folder:



For creating the folder and teacher.py:

Add_question.py

For Test file:

Do_test.py

For General Format:

Extract_path.py Import_file.py

For Doctest:

Test.py

For Random test:

Random_input_generator.py Random_strings_Property.py Add.py Diff.py

For Property Test:

Property_input_generator.py Tester.py

For students and teacher's function:

Add2ints folder

How to set up and use the demo (example add2ints):

The final demo here first tests the question statements, extra tests that professors provided in doc-string. It will then add the appropriate random tests based on the number of tests provided by the professors. Then, it will test the property test professor provided in doc-string and provided the output to each student.

Here are the steps:

1. Set up the current environment where the add_question.py is located. (In figure before, Demo would be the current position)

2. Professor can use the comment: python3 add_question.py function_name to create a folder with same function name and one teacher.py in it.
    a. Do not change the name of folder and teacher.py.
    b. Fill everything in teacher.py.

```
# teacher.py

# put parameters in the function definition; use type annotations, e.g.
# def add2lists(a:int, b:int):

def add2lists(x):
    """
    @Question statement
    (describe question here; include some doctests)

    @Extra Tests
    (optional doctests, not seen by student until they get marking back)

    @Properties
    (optional)
    """
    # put correct solution code here
    pass
```

    c. Put parameters in the function definition; use type annotations.
    e.g., def add2ints (a:int, b:int):
    d. Make sure that the function to be tested has the same argument names as those defined in the properties.
    for example, if we have the property:

```
@Properties
add2ints(a, b) == add2ints(b, a)
add2ints(a, 0) == a
add2ints(0, b) == b
add2ints(a, -a) == 0
Commutative
"""
```

    Our function should be.

```
def add2ints(a:int, b:int):
```

e. Make sure that there are no extra spaces between @Properties and property test and quotation. See figure in part d.

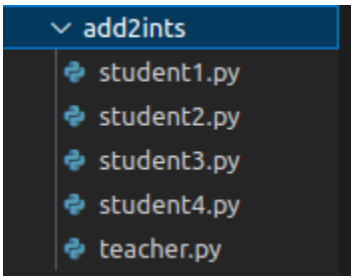
f. If do not want to add any properties or the extra tests, keep what the file gives before like.

```
@Properties
(optional)
"""
```

Example of filling out teacher.py:

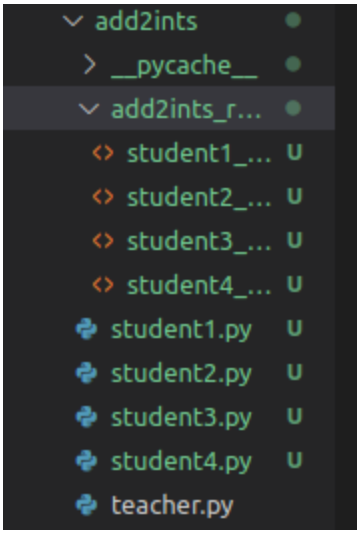
```
def add2ints(a:int, b:int):
    """
    @Question statement

    Write a function that takes two ints as input and returns their sum.

    >>> add2ints(4, 7)
    11
    >>> add2ints(-1, 3)
    2
    >>> add2ints(0, -5)
    -5

    @Extra Tests
    >>> add2ints(7, 8)
    15
    >>> add2ints(0, 0)
    0
    >>> add2ints(1, -1)
    0
    >>> add2ints(2, 2)
    4
    >>> add2ints(20, 14)
    34
    >>> add2ints(-26, 0)
    -26

    @Properties
    add2ints(a, b) == add2ints(b, a)
    add2ints(a, 0) == a
    add2ints(0, b) == b
    add2ints(a, -a) == 0
    Commutative
    """
    return a + b
```

3. Put students' work in folder we created before.



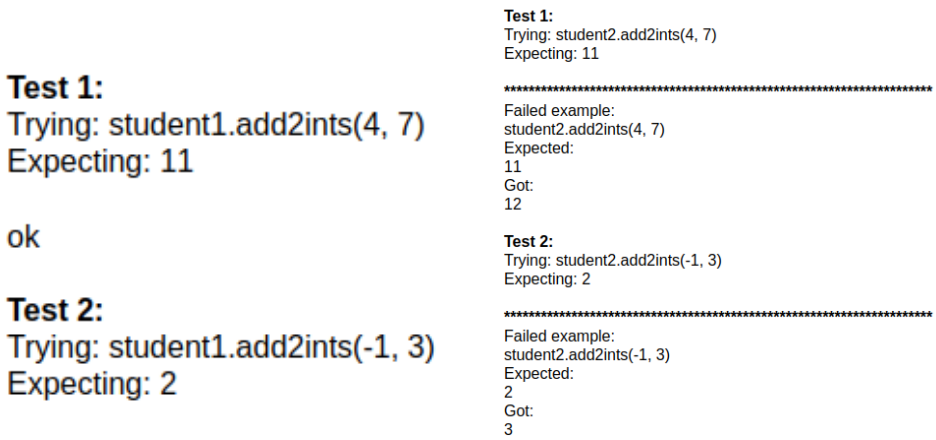
4. Professor can use the comment (same environment as before):
   python3 do_doctest.py function_name number_of_randomtest to test students' work.

5. Students' Output will be stored in the folder created before.



6. The output would be divided into two parts. The first part is about doctest, which gives the details about number of tests, true or false, the expected output, what students' function get and the score of the tests. The second part is property test, which gives 5 test cases, detailed of each property, results and the scores.

   For Doctest output:



Done Extra and Random Tests
19 of 19 tests passed (100% passed)

Done Extra and Random Tests
0 of 19 tests passed (0.0% passed)

For Property test output:

Running Property test

Test Cases	add2ints(a, b) == add2ints(b, a)	add2ints(a, 0) == a	add2ints(0, b) == b	add2ints(a, -a) == 0	Commutative	Pass Rate	Pass Percentage
a: -619 b: 778	True	True	True	True	True	5 of 5	100.0%
a: -75 b: 814	True	True	True	True	True	5 of 5	100.0%
a: 523 b: -508	True	True	True	True	True	5 of 5	100.0%
a: 568 b: -637	True	True	True	True	True	5 of 5	100.0%
a: -924 b: 101	True	True	True	True	True	5 of 5	100.0%

Running Property test

Test Cases	add2ints(a, b) == add2ints(b, a)	add2ints(a, 0) == a	add2ints(0, b) == b	add2ints(a, -a) == 0	Commutative	Pass Rate	Pass Percentage
a: -619 b: 778	True	False	False	False	True	2 of 5	40.0%
a: -75 b: 814	True	False	False	False	True	2 of 5	40.0%
a: 523 b: -508	True	False	False	False	True	2 of 5	40.0%
a: 568 b: -637	True	False	False	False	True	2 of 5	40.0%
a: -924 b: 101	True	False	False	False	True	2 of 5	40.0%