PA1

PA1

CSE 8A Fall 2021 PA1

Due date: Tuesday, October 5th @ 11:59PM PDT

(No late submission is allowed)

Provided Files

None

File(s) to Submit

- `pa1.py` (details on how to submit your file can be found below)
- You are also required to complete the **README** Quiz (20 points)

Part 0. Academic Integrity Agreement

Collaboration and Academic Integrity - Remember that the official academic integrity policy for the course is on the course syllabus.

We take Academic Integrity (AI) very seriously at UCSD. Before beginning the assignment, please complete the Academic Integrity Form. **You must fill this out in order to receive credit for any assignment.** Make sure to use your correct UCSD email when filling out the form.

Part 1. Implementation (80 points)

In this programming assignment, you will be asking the user for the coordinates of three points (the center point, followed by two points, each from the circumference of two concentric circles) to calculate the area of a ring shape and print the result.

Please click on **New File** button on the right panel (or click on the **+** button and select "New File"), and then enter `pa1.py` after `/home/`. You should now have a Python file called `pa1.py`. You are going to work on this file for the coding part of this assignment.

Part 1.1 Task Description

Your task is to write a program that takes in numeric inputs from the user representing the coordinates of three points, and prints the area of a ring shape derived from these three points.

A ring shape is basically composed of two concentric circles (i.e. two circles with the same center point). In this program, the user will:

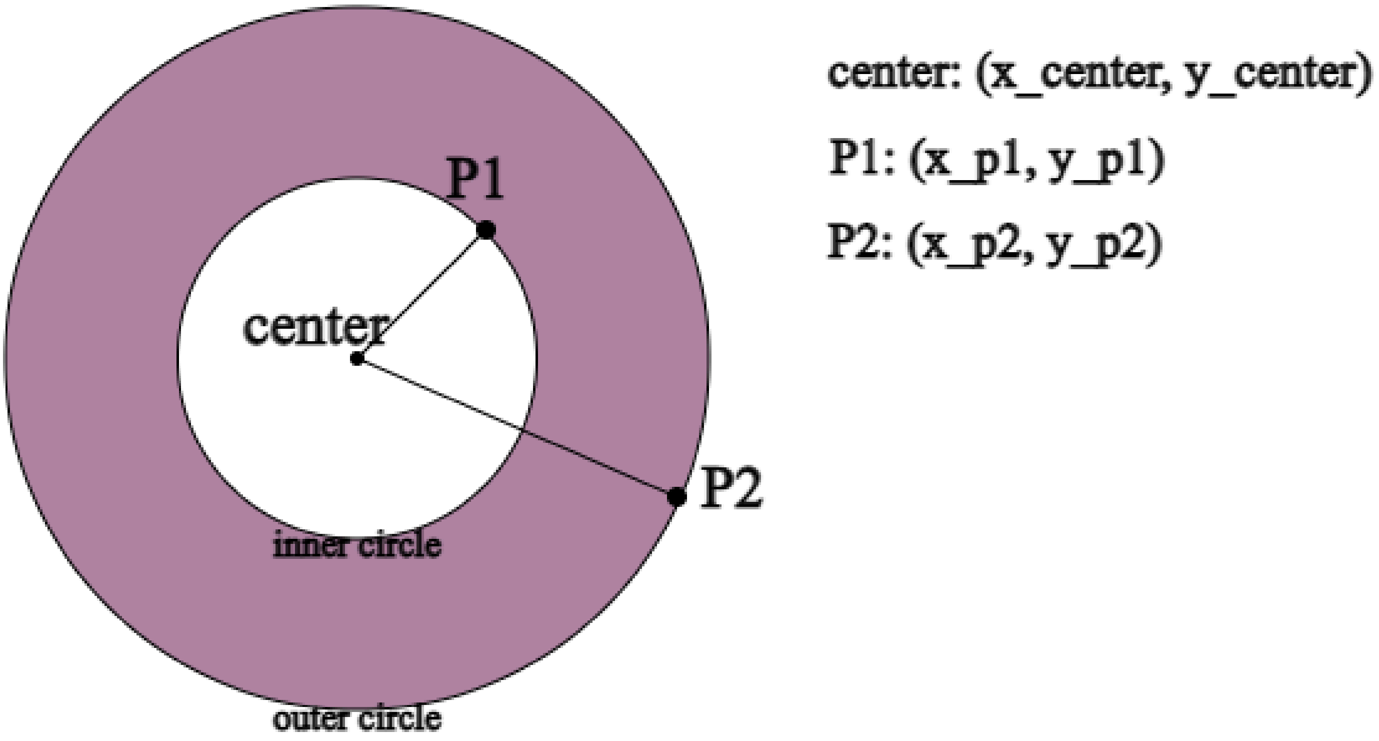
First input numbers `x_center` and `y_center` representing the **(x_center, y_center)** coordinates for the **center point**,

Then numbers `x_p1` and `y_p1` representing the **(x_p1, y_p1)** coordinates for **P1**, a point on one circle of the ring shape,

And lastly numbers `x_p2` and `y_p2` representing the **(x_p2, y_p2)** coordinates for **P2**, a point on the other circle of the ring shape.

Note that the first pair of input numbers is always the coordinates of the center, while the second and the third pairs of coordinates are input in an arbitrary order. With that being said, we don't know which point between P1 and P2 gives you the inner circle, and which one gives you the outer circle.

The graph below shows you the case when P1 resides on the inner circle and P2 resides on the outer circle, and the purple area is the area of the ring shape:



As you can see, the area of the ring shape is basically the difference between the area of the inner circle and the area of the outer circle.

You will use the following formula to compute the **area of a circle**:

$$area\ of\ circle\ =\ 3.14\ *\ \left((y\ -\ y_{center})^2 + (x\ -\ x_{center})^2\right)$$

where *x_center* and *y_center* are the xy-coordinates of the circle's center point, and *x* and *y* are the coordinates of a point on the circle's circumference. For the simplicity of testing, please use 3.14 as the value of `pi`.

Just for your curiosity, the steps below show you how the formula above was derived. **We are not asking you to implement these steps. You can use the formula above directly.**

$$area\ of\ circle\ =\ 3.14\ \star\ radius^2$$

The radius is the Euclidean distance between the center point and a point on the circle's circumference.

$$radius\ =\ \sqrt{(y\ -\ y_{center})^2\ +\ (x\ -\ x_{center})^2}$$

Now we can plug the radius into the area formula

$$area\ of\ circle\ =\ 3.14\ \star\ \left(\sqrt{(y\ -\ y_{center})^2\ +\ (x\ -\ x_{center})^2}\right)^2$$

The square root and square cancel which gives the formula

$$area\ of\ circle\ =\ 3.14\ *\ \left((y\ -\ y_{center})^2\ +\ (x\ -\ x_{center})^2\right)$$

After calculating the area of these two concentric circles, the **area of the ring shape** is the difference between them.

$$area\ of\ ring\ shape\ =\ area\ of\ outer\ circle\ -\ area\ of\ inner\ circle$$

The following steps show you some implementation details, and how you can run and test your program.

Part 1.2 Implementation Details & How to Run/Test Your Program

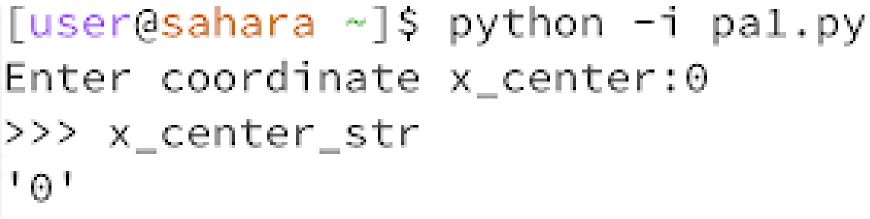
1. Your program should prompt the user with the messages "Enter coordinate x_center:", "Enter coordinate y_center:", "Enter coordinate x_p1:", "Enter coordinate y_p1:", "Enter coordinate x_p2:", "Enter coordinate y_p2:". In pa1.py, define variables called `x_center_str`, `y_center_str`, `x_p1_str`, `y_p1_str`, `x_p2_str`, `y_p2_str` that each contain a **string** received as input from the user (hint: use the **input()** function). You can start with only the first message: "Enter coordinate x_center:" and store the input in the variable `x_center_str`.

2. At the bottom of the screen there should be a button that says "Click here to activate the terminal". Click it and you should have a terminal running.

3. In the terminal, enter the command:

```
python -i pa1.py
```

The screenshot below shows what it should look like if everything is working (with taking in only the first coordinate `x_center_str`). Note two things: My program printed "Enter coordinate x_center:", then I typed in the 0 and pressed Enter. Then, at the Python prompt, I typed in the name of the `x_center_str` variable to check that it had the value I expected (in this case, the string "0").



```
[user@sahara ~]$ python -i pa1.py
Enter coordinate x_center:0
>>> x_center_str
'0'
```

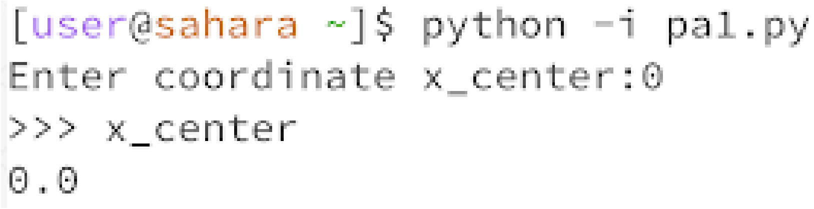
4. If your program doesn't have the behavior described above, re-check your program to see if you might have made a mistake – did Python give you an error message? Did you make a typo in the name of a variable or command?

5. Once the above interaction works, add another variable definition to your program after `x_center_str` called `x_center`. You should make it so your program stores the numeric value in

`x_center` represented by the string `x_center_str`. Hint: use the **float()** function!

- Repeat this for all of the strings (`x_center_str`, `y_center_str`, `x_p1_str`, `y_p1_str`, `x_p2_str`, `y_p2_str`)

6. Check that your program works before moving on – run the command: `python -i pa1.py`, give inputs, check that the variables have the values you expect. Below is an example for taking only input for the first coordinate.



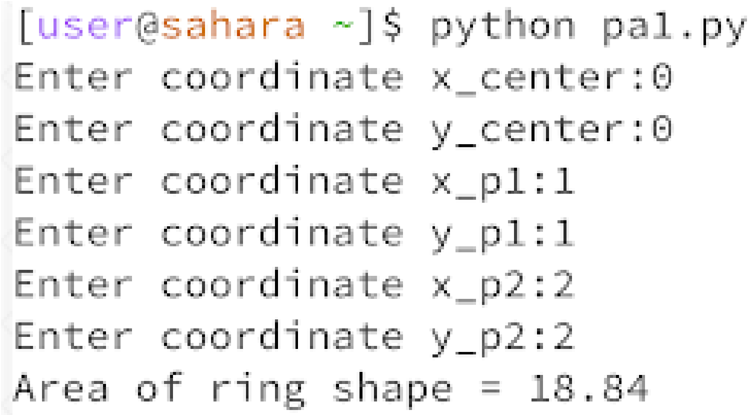
```
[user@sahara ~]$ python -i pa1.py
Enter coordinate x_center:0
>>> x_center
0.0
```

NOTE: You can use the **reset** button on the terminal or **exit()** to return to the terminal and run the command.

7. Once that part works, define another variable called `area_circle1` and `area_circle2` that contains the area of each circle computed from the input coordinates to the center. Run the program and check again that it works as expected.

8. Calculate the area of the ring shape (area of bigger circle - area of smaller circle). Since the second and third coordinates can come in any arbitrary order, we don't know which circle is larger (hint: use the **abs()** function). Since the calculation involves square of float numbers, you'll need to round your answer to **two decimal places** (hint: use the **round()** function)

9. Finally, add a print statement at the end of your program that displays the value of the area of the ring after the message. A run of your program should now look like this (note the lack of `-i`!). Now your program should ask you for input coordinates and it should produce an output area like below.

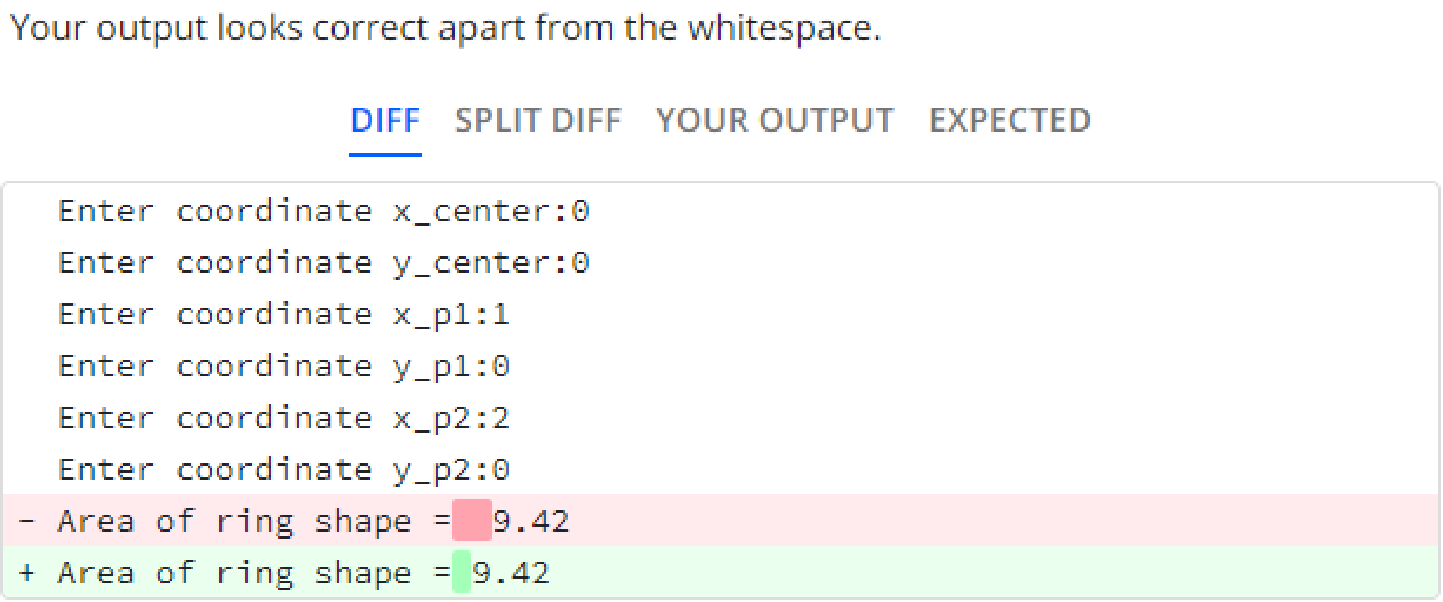


```
[user@sahara ~]$ python pa1.py
Enter coordinate x_center:0
Enter coordinate y_center:0
Enter coordinate x_p1:1
Enter coordinate y_p1:1
Enter coordinate x_p2:2
Enter coordinate y_p2:2
Area of ring shape = 18.84
```

Feel free to test your program with various inputs (different coordinate values). Once you are confident that your program is correct, you may submit it in the following way:

At the lower right part of the screen, you should see a **Mark** button:

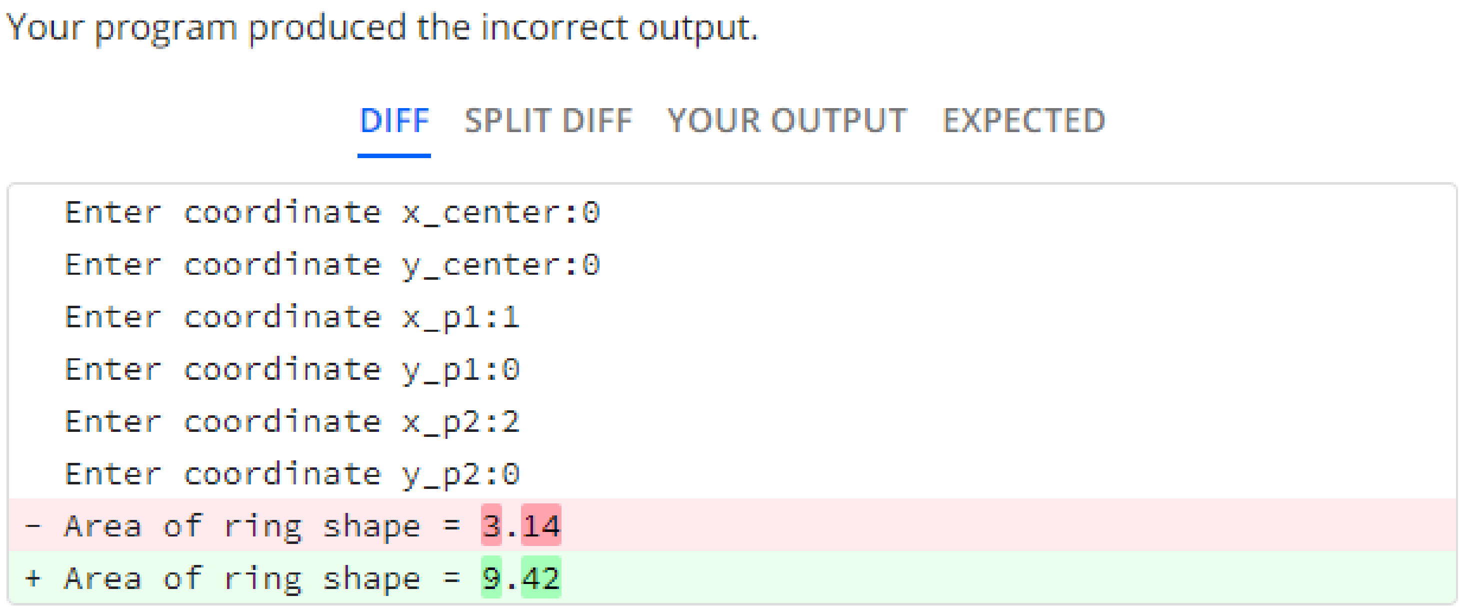
Click this button and you should be able to see the feedback for your program. If the feedback says "Passed", congratulations! You have finished the coding part of this assignment. If not, that means your program fails one or more test cases. For example, you may have feedback that looks like this:


Your output looks correct apart from the whitespace.

```
  Enter coordinate x_center:0
  Enter coordinate y_center:0
  Enter coordinate x_p1:1
  Enter coordinate y_p1:0
  Enter coordinate x_p2:2
  Enter coordinate y_p2:0
- Area of ring shape =  9.42
+ Area of ring shape = 9.42
```

It means you may have accidentally put an additional whitespace after '='.

Or you may have feedback that looks like this:

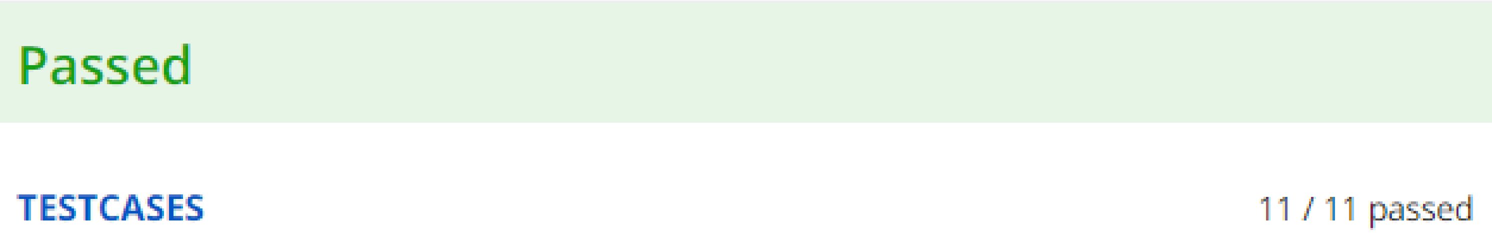

Your program produced the incorrect output.

```
  Enter coordinate x_center:0
  Enter coordinate y_center:0
  Enter coordinate x_p1:1
  Enter coordinate y_p1:0
  Enter coordinate x_p2:2
  Enter coordinate y_p2:0
- Area of ring shape = 3.14
+ Area of ring shape = 9.42
```

It could mean that your program isn't doing the right calculation. Are you using the correct formula?

But don't worry too much as you can submit multiple times! You may continue submitting until your program passes all test cases.

For this PA, we have **11 test cases** in total. If you fail some tests among the 11 test cases, you might see a score out of a number less than 11 (e.g. 0/1 passed, 7/8 passed). This is normal as tests are executed in order, and failing one test stops the autograder from executing the rest of the test cases. When you pass all the test cases, you should be able to see this:



Passed

TESTCASES

11 / 11 passed

Part 2: README Quiz (20 points)

You are required to complete the README in the PA1 lesson. There is no time limit on the README but you must submit the README by the PA deadline. You can submit multiple times before the deadline. Your latest submission will be graded.

Submission

Turning in your code

- Submit `pa1.py` to PA1 Lesson on EdStem via the "Mark" Button by **Tuesday, October 5th @ 11:59PM PDT. No late submission is allowed.**

- Also, submit README Quiz by the above deadline

Evaluation

- Implementation Correctness (80 points): You will earn points based on the autograder tests that your code passes. If the autograder tests are not able to run (e.g., your code does not compile or it does not match the specifications in this writeup), you may not earn credit.
- README Quiz (20 points)

README

Question 1 *Submitted Sep 30th 2021 at 10:16:00 am*

True or False: Python does not make programmers specify a type when declaring a variable but we can choose to do so.

● True

○ False

Question 2 *Submitted Sep 30th 2021 at 10:16:12 am*

What is the value of previous_year when the following code is executed?

```
year = 2021
previous_year = year - 1
year = 2022
previous_year
```

● 2020

○ 2021

○ 2022

○ error

Question 3 *Submitted Sep 30th 2021 at 10:20:40 am*

Which of the following rules are true about policies governing programming assignments for CSE 8A? Select all that apply

□ After a student received the grade of an assignment, she has 7 days to resubmit modified codes and may get up to 50% of lost points for the entire assignment.

☑ After a student received the grade of an assignment, she has 7 days to resubmit modified codes and may get up to 50% of lost points for auto-graded part of the assignment.

☑ No late work is accepted unless it is due to documented emergencies.

☐ You cannot request a regrade for PAs. Everything is set to stone after the grade is released.

Question 4 *Submitted Sep 30th 2021 at 10:19:49 am*

Which of the following rules are true about policies governing labs for CSE 8A? Select all that apply

☐ Labs happen on Thursday and you don't have to attend any lab session.

☑ You should go to the lab session that you are enrolled in.

☑ Each lab has lab activities (70% of overall grade for that lab) and lab quiz (30% of overall grade for that lab)

☐ Lab quizzes are open to collaborations

Question 5 *Submitted Sep 30th 2021 at 10:20:06 am*

Which of the following are true about academic integrity policy for CSE 8A? Select all that apply

☑ Students are allowed to ask teaching staff any questions about programming assignments and lab.

☐ Students are allowed to share their code with each other for some assignments

☑ Students are allowed to discuss general strategies to solve programming assignments but are not allowed to discuss algorithms in detail.

☑ Students can collaborate on lab activities but not on lab quizzes.

☐ Students are allowed to share their codes after the deadlines of that assignment have passed.