
Programming Problem 6 - parabola.py

Directions: Download the template files I have provided on Blackboard. Then open Spyder, load the template file, and write the following program. Submit your source code via Gradescope in .py format; do NOT send any other files. READ THE INSTRUCTIONS on how to submit your work in the Course Documents section of Blackboard.

Be sure to read the SPECIFICATIONS carefully! And write comments!

Recall that the graph of a quadratic equation $y = ax^2 + bx + c$, $a \neq 0$ is a parabola. The equation can be written in *vertex form* as $y = a(x - h)^2 + k$, where (h, k) is the vertex of the parabola.

Create a class definition for a class called `Parabola`.

The objects of this class have the following attributes:

- `._xvert`, representing the x-coordinate of the vertex;
- `._yvert`, representing the y-coordinate of the vertex;
- `._leadingcoef`, representing the leading coefficient (a) in the equation for the parabola;

The class definition should also contain the following methods:

- a constructor (that's another name for the `.__init__()` function), which takes 3 floats, which set the attribute variables in the order given above;
- `.yintercept()`, which takes no outside arguments, and **returns** (NOT prints) the y-coordinate of the y-intercept of the parabola (recall you can find the y-intercept by setting x equal to zero in the equation for the parabola);
- `.display()`, which takes no outside arguments and returns nothing, but prints out the attribute variables (just print out the attribute variables in one print statement, in the order shown above, and nothing else);
- `.isConcaveDown()` which takes no outside arguments, and returns `True` if the parabola is concave down (that is, if $a < 0$) and `False` otherwise.

Your class definition should make my client code work when uploaded to Gradescope and the test cases are run. (You won't see anything when you run the code on Spyder, unless you read through my template code to see what it is doing and know what needs to be input.)

You should also add code in `main()` which uses `.display()` and `.yintercept()` to both display parabola information and to print the y-intercepts of all `Parabolas` in the list `my_parabolas` that are concave down. This code should not have any direct references to `._leadingcoef`, `._xvert`, or `._yvert`.

For example, if I had the following code in `main()`:

```
para1 = Parabola(1.1, 1.2, 5.1)
para2 = Parabola(-1.5, 2.2, -3.5)
my_parabolas = [para1, para2]
```

Then your client code should print:

```
-1.5 2.2 -3.5
-5.675
```

Since `para2` is the only `Parabola` in `my_parabolas` that is concave down, and its vertex is at $(-1.5, 2.2)$, its leading coefficient is -3.5 , and its y-intercept is -5.675 .

Specifications: your program must

- not modify any of the starter code in the template provided
- contain a definition of the class `Parabola` that includes the methods and attributes listed above, and which allows my test code to run properly.
- contain client code in `main()` which use `.display()` to display the attributes of the `Parabolas` in the list `my_parabolas` that are concave down.
- not contain any direct references to attribute variables in the client code in `main()`. **Note that you may pass the test cases on Gradescope even if you violate this specification, but your points will then be manually deducted later.**