

CSE 30

Programming Abstractions: Python

Programming Assignment 3

In this assignment you will create a class called `Lines` that will be placed in a file called `lines.py`. Much of the work is already done in the template file `lines_stub.py`, posted at

https://classes.soe.ucsc.edu/cse030/Spring21/Examples/pa3/lines_stub.py

The `Line` class provides an abstract representation of a line in the x - y coordinate plane. Included in the above stub is the `Point` class we covered in the `geometry.py` example, also posted on the class webpage. Your task is fill in the definitions of the methods (10 in all) belonging to the `Line` class, and to add one new method to the `Point` class.

The `Line` class represents a line in the plane by maintaining two attributes: a `Point` object representing a point on the line, and a numeric value giving its slope. The obvious names for these attributes are *point* and *slope* respectively, which are used in this document, but you can give them any names you like. One logical problem entailed by this representation is how to represent vertical lines, i.e., lines that have no numerical value as slope. Such lines are often said to have "infinite slope" in Geometry textbooks. We will handle this situation by simply setting the *slope* attribute, normally a number, to be the string `'infinity'`. This choice will create some special cases in the design of several of the required functions, and it will be your task to sort out these cases logically.

This project requires some familiarity with basic facts of plane geometry, but nothing more than would be encountered in a class like MATH 3 (precalculus algebra), a prerequisite for this course. For instance, the function `parallel()` in the `Line` class, when called as `A.parallel(B)`, returns `True` if line *A* is parallel to line *B*, and `False` otherwise. But as is well known, two lines are parallel if and only if they have the same slope. Another example is `intersect()`, contained in the `Line` class. The call `A.intersect(B)` will return the `Point` object representing the point of intersection of line *A* with line *B*. This is a common homework problem in elementary algebra, i.e., find the point of intersection of two lines whose equations are given. You must solve the general problem: find formulas for the coordinates of the intersection point, and account for the special cases where the lines are parallel (return `None`), or where one line is vertical.

The correct output for function `main()` is included as a comment in the above template. If your function definitions are correct, the output will match the quoted output exactly. However, `main()` should not be considered a complete test of all of the required functions. As always, you should thoroughly test every logical pathway within every function you design, making sure that it strictly adheres to its specification. To do this testing, you may alter `main()` temporarily, but be sure to return it to its original state before you submit the file.

Once all coding and testing are complete, change the name of the file from `lines_stub.py` to `lines.py` and submit it to Gradescope before the due date. As always, ask questions and get help early.