

Assignment 07: Object-Oriented Programming

Due date: Wednesday, 18 March, 12:00pm

Question 1: Turtle Graphics

Logo is one of the oldest programming languages, dating back to 1967. At its core, Logo is a language for creating drawings based on *turtle graphics*. In turtle graphics, an object called the turtle sits in the plane. It has a few simple attributes:

- A current (x, y) position
- A current orientation (an angle between 0 and 2π)
- A pen, which can be either “up” (not drawing) or “down” (drawing)

At the start of a Logo program, the turtle sits in the middle of the screen, pointing up to the top.

The Logo language includes a number of built-in commands that manipulate the turtle, causing it to move, turn, and draw. Here is the subset of the language that we will deal with:

- FD 100: Move the turtle forward by 100 pixels in the direction it’s currently facing. If the pen is currently down, draw a line from the old turtle position to the new one.
- BK 85: Move the turtle backward by 85 pixels (i.e., opposite to the direction it’s facing). Don’t change the turtle’s orientation. As above, if the pen is down, draw a line.
- LT 50: Turn the turtle left (counterclockwise) by 50 degrees.
- RT 45: Turn the turtle right (clockwise) by 45 degrees.
- PU: Pen up—lift the pen off the drawing. Future commands won’t leave lines behind.
- PD: Pen down—place the pen on the drawing. Future commands will leave lines behind.
- RESET: Clear the drawing.

Obviously, the numbers in the lines of code above are just examples; you can move by any distance and turn by any angle.

Download the starter code in `A07Logo.zip`. The code comes with a `Point` class like the one discussed in lectures, and some other functions that use `ControlP5` and a bit of Processing trickery to set up the rest of the interpreter. However, the starter code won’t run out of the box, because it assumes the existence of a `Turtle` class that implements methods for the Logo commands above. Your task is to complete the interpreter by writing the `Turtle` class. Your class will need to contain the following:

- The “few simple attributes” mentioned above. Be sure to choose appropriate types for these three fields.
- A default constructor (i.e., a constructor that takes zero arguments). The constructor should set the turtle to position $(0, 0)$, angle $-\pi/2$ (which will make it point up) and pen enabled.
- Methods `forward()` and `backward()` that move the turtle as described above. These methods each take a single `float` argument for the distance to move by.
- Methods `left()` and `right()` that rotate the turtle as described above. These methods each take a single `float` argument representing an angular difference *in degrees*. You’ll want to convert to radians in these methods.
- Methods `penDown()` and `penUp()` that tell the turtle whether to draw a line when it moves.
- A `draw()` method with no arguments. This method draws a representation of the turtle itself at its current location, pointing in its current orientation.

The rest of the provided code sets up a text field at the bottom of the window. You can type Logo commands like the ones above into the field, and they’ll be executed one at a time. I’ve also provided a command of the form `run some_file.txt` that loads a file containing Logo commands and executes them all in sequence. That could be useful for writing a longer test drawing and running it repeatedly without having to type it in each time. You shouldn’t need to write any code outside the `Turtle` class.

What to submit: On LEARN, you should submit the sketch `A07Logo`. Submit the entire sketch folder.