CS116 – Objects, Methods, and Parameter Passing

Purpose: A popular pattern in computer programming is to represent logical constructs as *objects* that model interaction with other objects as they might in the real world. The sort of interaction a particular object is capable of is described in the *class* that defines it. Therefore, an object is a specific instance of a class definition. The kinds of actions an object is capable of, (described in the class definition) are called *methods*. Let's say we have a class defining the behavior of turtles. A turtle might move in a given direction, or turn to face a new direction. A turtle might have a name. It might occupy a position on an x/y plane (that plane would likely have its own class definition).

Skills: After completion of this module you should be able to

1. Create objects from classes
2. Use a method on an object
3. Add a new method to a class
4. Use a reference for a class API which describes the methods

Knowledge: This module will help you become familiar with the following content knowledge:

1. Classes and Objects
2. Parameter Passing

Activity: With your group perform the following tasks and answer the questions. You will be reporting your answers back to the class in 45 minutes.

1. Type the following in the console to create and use a Picture object called p.

   ```
   p = Picture(500,300)
   p.show()
   ```

   Create a second Picture object called q and use the method show on it.

   ```
   q = Picture(100,200)
   q.show()
   ```

   What do the parameter values (500,300) and (100,200) do when we create these objects?

   How would you create another blank picture which is square in its dimensions with each side of size 400?

2. Create a Sound object called s and use the play method on it. You will be given a chooser to select a wav file and can use any that have been provided in the course materials. The file name that you select is provided as the parameter value.

   ```
   s = Sound(pickAFile())
   s.play()
   ```

   How would you create a Picture object from a file that you can select?

   How would you create a Text object from a file that you select and then use a method called getWords to get a list of words in the file?

3. Create a Turtle object called t in the console. Now take a look at the code for the `drawSquare` method included in the myturtle.py file.

   ```
   def drawSquare(self,width):
       self.forward(width)
       self.right(90)
       self.forward(width)
       self.right(90)
       self.forward(width)
       self.right(90)
       self.forward(width)
       self.right(90)
   ```

   Use this method on a Turtle object in the console to draw a square of size 100 by typing

   ```
   t.drawSquare(100)
   ```

   Have the turtle draw a second square of size 50.
   How does the parameter `width` get its value?

   What would be the value of `width` in your first use of the `drawSquare` method?

   What would be the value of `width` in the second use the `drawSquare` method?

4. Create a second Turtle object called s in the world and position the turtle at location (100,50) by typing

   ```
   s = Turtle(100,50)
   ```

   Make s draw a square of size 50.
   In the code for `drawSquare` we are using Turtle methods on an object called `self`. What is the object that `self` refers to in the use of `drawSquare` that you just performed?

   What is the object that `self` refers to in your very first use of `drawSquare`?

   How is `self` getting its value?

   Explain how each of the eight method uses (we will call these ”invocations”) in the `drawSquare` method definition causes a square to be drawn by the turtle.

5. Now take a look at the code for the `drawNestedSquares` method included in the myturtle.py file but don't trying using this method yet.

   ```
   def drawNestedSquares(self):
       self.drawSquare(50)
       self.drawSquare(100)
       self.drawSquare(150)
   ```

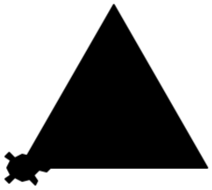
   This method uses the `drawSquare` method that we just used. Try to predict what this code will do.

   Now use this method with one of your Turtle objects. If it did not perform as you expected then try to resolve any differences and explain them.

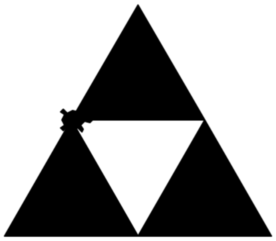
Complete each of the following assignments to be submitted for grading. Each should be done individually but you can consult with a classmate to discuss your strategies or if you get an error message that you do not understand.

Assignment 1:
Write an additional method for the Turtle class which has the turtle draw a triangle where each side is the same size and this length is specified by a parameter. The interior angles in a triangle will all be 60 degrees. To fill in the triangle you will want to look at the `begin_fill` and `end_fill` methods.

Criteria for Success: The triangle should look like this



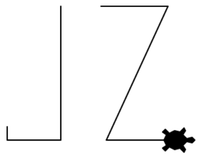
Assignment 2:
Write a method which draws three triangles as pictured.



Criteria for Success: Your new method should invoke the `drawTriangle` method three times.

Assignment 3:
The methods available for Turtles is provided in your course documentation. Use some of these methods to write a method `myInitials` which has the Turtle draw your initials.

Criteria for Success: You have your initials drawn on the screen. Here are mine!



Submit your modified myturtle.py file with your methods in Canvas for grading.