CS/IT 102: Homework 11 (60 points)

Due Date: **5:00 p.m. <u>Sunday May 5th</u>**

Submission

- Reminder: This is an INDIVIDUAL ASSIGNMENT. You may discuss the assignment, but you may not share code.
- Submit the file to the Homework 11 dropbox on Kodiak.
- Make sure that each file, each class and each method in a class has a comment header

Grading:

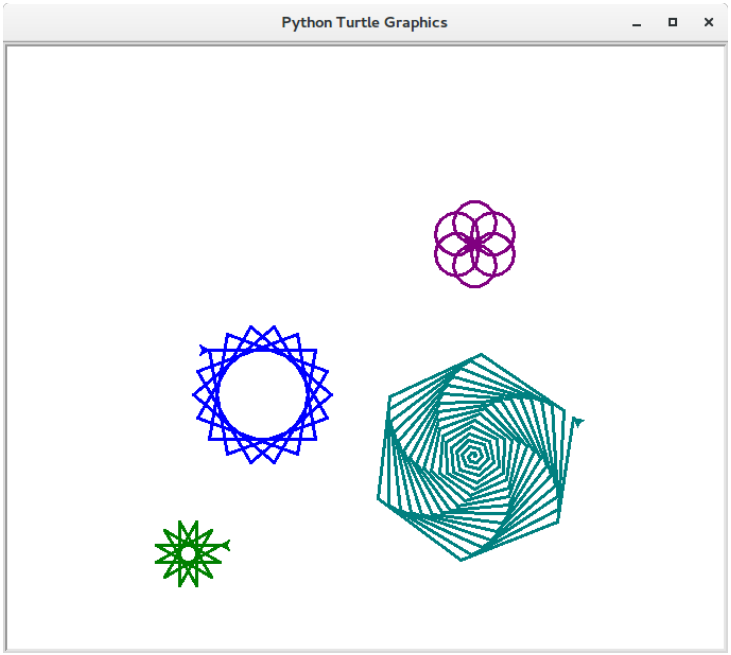
- *Class TurtleGraphic: **24 points***
  - *Questions 1, 2, 3, and 6: **3 points each***
  - *Questions 4 and 5: **6 points each***
- *Wheel class that inherits from TurtleGraphic: **10 points***
- *Your own designed class that inherits from TurtleGraphic: **10 points***
- *Program runs and is bug free: **10 points***
- *Code commenting and variables naming: **6 points***

Domain: We will continue to explore the turtle graphics domain. As you may have noticed, the `Wreath`, `Flower`, and `Wheel` classes used in Homework 10 have a lot of identical code in them. As part of this lab, you must create a super class called `TurtleGraphic` and put all of the common code in this class. Specifically, you must do the following:

1. Create a `TurtleGraphic` class in a file called `turtlegraphic.py` that contains the following methods:
   1. `__init__(self, turt, name, color, speed, xcoord, ycoord)` - note that this code is already contained in the `__init__` method of individual `Flower` and `Wheel` classes from homework 10.
   2. `def draw(self)` - sets the turtles speed, pen size, and color and moves the turtle to the correct starting point. Note that this code is already the first seven lines in the `draw` method of individual `Flower` and `Wheel` classes from homework 10.
   3. `def __str__(self)` - note that this code is already contained in the `__str__` method of individual `Flower` and `Wheel` classes from homework 10.
   4. Setters/Mutators for the following attributes: `color`, `xcoord`, `ycoord`, and `speed`. Note that these methods are already contained in the individual `Flower` and `Wheel` classes from homework 10.
   5. Getter/Accessors for the following attributes: `color`, `xcoord`, `ycoord`, and `speed`. Note that these methods are already contained in the individual `Flower` and `Wheel` classes from homework 10.
   6. Add a "getter" method to retrieve the turtle. This method is needed by the subtypes to draw the individual graphic.
2. We have provided you with a `Flower` and a `Wreath` class that inherit from the `TurtleGraphic` class. You must create a similar `Wheel` class that also inherits from the `TurtleGraphic` class and draws a wheel on the screen.
3. Create one more class of your own design. In the example below, we have created a `Spiral` class. You may create a simple class such as a square, or any other graphic that you like. This class MUST inherit from `TurtleGraphic`.

Hints:

- A `main.py` file has been provided. Your code should run with this file. An example run would appear as:



- Each class should be stored in a separate file.