**Outline**

Develop a better understanding of procedural sequencing by solving shape drawing challenges using the turtle environment.

**Objectives**

* Use correct terminology to describe programming concepts;
* Describe the types of data that computers can process and store (e.g., numbers, text);
* Explain the difference between constants and variables used in programming;
* Use variables, expressions, and assignment statements to store and manipulate numbers and text in a program

**Materials**

* Python Turtle Development Environment at: https://repl.it/
* PythonWorksheetII form the GitHub Repository
* Web links identified in the questions below

**Level 1: Drawing Basic Shapes With Python Turtle**

1. Open the document PythonWorksheetII from the class GItHub repository. Read over “Part III” at the end of the PythonWorksheetII document.
2. Create an new Repl by selecting the “Python with Turtle” language / environment.
3. Begin all of your turtle programs with the following code to create a “pen”:

import turtle

myPen = turtle.Turtle()

1. Create a program to draw a red circle.
   1. Demonstrate your program to Mr. Nestor
   2. Provide a listing of your program code below:
2. import turtle
3. myPen = turtle.Turtle()
4. myPen.color("red")
5. myPen.circle(60)
6. Create a program to draw any three of the shapes described in “Part III” of   
   the PythonWorksheetII document.
   1. Demonstrate your programs to Mr. Nestor
   2. Provide a listing of your program code below:

import turtle

myPen = turtle.Turtle()

myPen.color("blue")

myPen.forward(100)

myPen.left(91)

myPen.forward(100)

myPen.left(91)

myPen.forward(100)

myPen.left(91)

myPen.forward(100)

**Level 2: Filled Shapes & Spirals**

1. Review the sample code for creating filled shapes at:   
   <http://www.pythoncode.co.uk/turtle-challenge-3>.
2. Complete the challenge described at: <http://www.pythoncode.co.uk/turtle-challenge-4>
   1. Demonstrate your programs to Mr. Nestor
   2. Provide a listing of your program code below:

import turtle

myPen = turtle.Turtle()

myPen.color("red")

myPen.begin\_fill()

myPen.forward(100)

myPen.left(90)

myPen.forward(100)

myPen.left(90)

myPen.forward(100)

myPen.left(90)

myPen.forward(100)

myPen.end\_fill()

myPen.forward(100)

myPen.left(90)

myPen.forward(100)

myPen.left(90)

myPen.forward(100)

myPen.left(90)

myPen.forward(100)

myPen.forward(100)

myPen.left(90)

myPen.forward(100)

myPen.left(90)

myPen.forward(100)

myPen.left(90)

myPen.forward(100)

myPen.forward(100)

myPen.left(90)

myPen.forward(100)

myPen.left(90)

myPen.forward(100)

myPen.left(90)

myPen.forward(100)

1. Review the sample code for creating filled shapes at:   
   <http://www.pythoncode.co.uk/turtle-challenge-5>
2. Complete the challenge described at: <http://www.pythoncode.co.uk/turtle-challenge-6>
   1. Demonstrate your programs to Mr. Nestor
   2. Provide a listing of your program code below:
3. import turtle
4. myPen = turtle.Turtle()
5. myPen.speed(0)
6. myPen.color("blue")
7. side=400
8. myPen.penup()
9. myPen.pendown()
10. for i in range (1,100):
11. myPen.forward(side)
12. myPen.left(90)
13. side=side-4

**Level 3: Four Quadrant Cross Challenge**

1. Complete the challenge described at: <http://www.101computing.net/python-turtle-challenge/>
2. Demonstrate your programs to Mr. Nestor
3. Provide a listing of your program code below:
4. import turtle
5. myPen = turtle.Turtle()
6. myPen.shape("arrow")
7. myPen.color("purple")
8. myPen.delay(5)
9. for i in range(0,11):
10. yFrom=10-i
11. xTo=i
12. myPen.penup()
13. myPen.goto(0,20\*yFrom)
14. myPen.pendown()
15. myPen.goto(20\*xTo,0)
16. myPen.left(180)
17. myPen.forward(200)
18. for i in range(0,11):
19. yFrom=10-i
20. xTo=i
21. myPen.penup()
22. myPen.goto(0,+20\*yFrom)
23. myPen.pendown()
24. myPen.goto(-20\*xTo,0)
25. for i in range(0,11):
26. yFrom=-10+i
27. xTo=i
28. myPen.penup()
29. myPen.goto(0,20\*yFrom)
30. myPen.pendown()
31. myPen.goto(20\*xTo,0)
33. for i in range(0,11):
34. yFrom=10-i
35. xTo=i
36. myPen.penup()
37. myPen.goto(0,-20\*yFrom)
38. myPen.pendown()
39. myPen.goto(-20\*xTo,0)