**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**

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

import turtle

myPen = turtle.Turtle()

* Create a program to draw a red circle.
* Demonstrate your program to Mr. Nestor
* Provide a listing of your program code below:
* Create a program to draw any three of the shapes described in “Part III” of   
  the PythonWorksheetII document.
* Demonstrate your programs to Mr. Nestor
* Provide a listing of your program code below:

**Level 2: Filled Shapes & Spirals**

* Review the sample code for creating filled shapes at:   
  <http://www.pythoncode.co.uk/turtle-challenge-3>.

import turtle

myPen = turtle.Turtle()

myPen.forward(120)

myPen.right(60)

myPen.forward(120)

myPen.right(60)

myPen.forward(120)

myPen.right(60)

myPen.forward(120)

myPen.right(60)

myPen.forward(120)

myPen.right(60)

myPen.forward(120)

* Complete the challenge described at: <http://www.pythoncode.co.uk/turtle-challenge-4>
* Demonstrate your programs to Mr. Nestor
* Provide a listing of your program code below:

myPen.right(90)

myPen.forward(100)

myPen.right(90)

myPen.forward(50)

myPen.right(90)

myPen.forward(50)

myPen.right(90)

myPen.forward(100)

myPen.right(90)

myPen.forward(50)

myPen.right(90)

myPen.forward(50)

myPen.right(90)

myPen.forward(1)

myPen.left(90)

myPen.forward(1)

myPen.forward(50)

myPen.right(90)

myPen.forward(50)

myPen.right(90)

myPen.forward(3)

myPen.right(90)

myPen.forward(50)

myPen.begin\_fill()

myPen.left(90)

myPen.forward(50)

myPen.left(90)

myPen.forward(50)

myPen.left(90)

myPen.forward(50)

myPen.end\_fill()

* Review the sample code for creating filled shapes at:   
  <http://www.pythoncode.co.uk/turtle-challenge-5>

import turtle

myPen = turtle.Turtle()

myPen.circle(120)

myPen.circle(110,)

myPen.circle(100,)

myPen.circle(90,)

myPen.circle(80,)

myPen.circle(70,)

myPen.circle(60,)

myPen.circle(50,)

myPen.circle(40,)

myPen.circle(30,)

myPen.circle(20,)

myPen.circle(10,)

* Complete the challenge described at: <http://www.pythoncode.co.uk/turtle-challenge-6>
* Demonstrate your programs to Mr. Nestor
* Provide a listing of your program code below:

import turtle

pen = turtle.Turtle()

pen.speed("fastest")

def spiral (expandSize):

pen.color("black")

lastLine = 0

timer = 0

size = expandSize

while (timer <= 25):

lastLine += 1

pen.left(90)

pen.fd(size)

if (lastLine == 1 or lastLine == 2):

size += 10

elif (lastLine == 3):

size += 10

lastLine = 0

timer += 1

spiral(50)

**Level 3: Four Quadrant Cross Challenge**

* Complete the challenge described at: <http://www.101computing.net/python-turtle-challenge/>
* Demonstrate your programs to Mr. Nestor
* Provide a listing of your program code below:

mport turtle

myPen = turtle.Turtle()

myPen.speed(100)

myPen.shape("arrow")

myPen.color("red")

myPen.delay(1) #Set the speed of the turtle

for i in range(0,11):

yFrom=10-i

xTo=i

myPen.penup()

myPen.goto(0,20\*yFrom)

myPen.pendown()

myPen.goto(20\*xTo,0)

for i in range(0,11):

yFrom=10-i

xTo=i

myPen.penup()

myPen.goto(0,-20\*yFrom)

myPen.pendown()

myPen.goto(20\*xTo,0)

for i in range(0,11):

yFrom=10-i

xTo=i

myPen.penup()

myPen.goto(-0,-20\*yFrom)

myPen.pendown()

myPen.goto(-20\*xTo,-0)

for i in range(0,11):

yFrom=10-i

xTo=i

myPen.penup()

myPen.goto(-0,20\*yFrom)

myPen.pendown()

myPen.goto(-20\*xTo,-0)