**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.  
   Done
2. Create an new Repl by selecting the “Python with Turtle” language / environment.

Done

1. 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. Provide a listing of your program code below:

import turtle

myPen = turtle.Turtle()

myPen.color("red")

myPen.circle(80)

1. Create a program to draw any three of the shapes described in “Part III” of   
   the PythonWorksheetII document.
   1. Provide a listing of your program code below:

Square:

myPen.forward(60)

myPen.left(90)

myPen.forward(60)

myPen.left(90)

myPen.forward(60)

myPen.left(90)

myPen.forward(60)

Plus:

myPen.forward(60)

myPen.left(90)

myPen.forward(30)

myPen.left(90)

myPen.forward(60)

myPen.right(90)

myPen.forward(60)

myPen.left(90)

myPen.forward(30)

myPen.left(90)

myPen.forward(60)

myPen.right(90)

myPen.forward(60)

myPen.left(90)

myPen.forward(30)

myPen.left(90)

myPen.forward(60)

myPen.right(90)

myPen.forward(60)

myPen.left(90)

myPen.forward(30)

myPen.left(90)

myPen.forward(60)

Square with circle in it:

myPen.color("red")

myPen.forward(100)

myPen.left(90)

myPen.forward(100)

myPen.left(90)

myPen.forward(100)

myPen.left(90)

myPen.forward(100)

myPen.color("blue")

myPen.up()

myPen.goto(0,50)

myPen.down()

for I in range(10):

myPen.circle(50)

myPen.color("blue")

myPen.circle(50)

**Level 2: Using a Loop**

1. Google the keywords “Python Turtle Methods”.
   1. Explain how the “goto” method works and how you could use it when drawing repeated shapes.

Goto is a way of moving your pen to a certain place without having to go left and right and turning it into a longer process.

* 1. List some other useful methods not listed in “Part III” at the end of the PythonWorksheetII document.

-heading None Returns the current heading

-position None Returns the current position

-begin\_fill None Remember the starting point for a filled polygon

-end\_fill None Close the polygon and fill with the current fill color

-dot None Leave a dot at the current position

stamp None Leaves an impression of a turtle shape at the current location

1. Create a repeating pattern on your screen. The pattern must meet the following requirements:
   1. The basic pattern must be made up of several individual Turtle methods (e.g. changes of colour, changes of direction, size, motion, etc.)
   2. The basic pattern must be repeated several times with a shift in starting position each time.
2. Use a Python Loop to create your repeating pattern
   1. The Loop may be a Counted Loop or a Conditional Loop
   2. The indented block of code for the loop should be your basic pattern.
3. Provide a listing of your repeating pattern loop below.

**Level 3: Defining a Function**

Convince and flexibility

1. Google the keywords “Python Function Syntax”.
   1. Explain what the “def” keyword does
   2. Explain any special rules regarding the function name
   3. Explain what the parameters (or arguments) do
   4. Where should the colon “:” be placed
   5. Explain how to write Python statements that make up the function body
   6. Explain the “return” statement
2. Provide an example of a simple function that uses one or more parameters.
   1. Write the function definition below
   2. Write some code to call the function below
3. Convert your basic pattern (from Level 2 above) into a function
4. The function name should be “my\_pattern”
5. The parameters should be the x and y starting position for your pattern
6. Your function does not need to use the “return” statement
7. Use a your basic pattern function and a Python Loop to create your repeating pattern
   1. The Loop may be a Counted Loop or a Conditional Loop
   2. Your function should be called from within the loop.
8. Provide a listing of your function definition and repeating pattern loop below.