**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()

4. Create a program to draw a red circle.

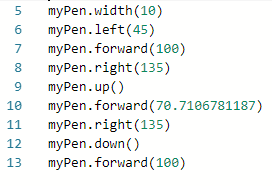
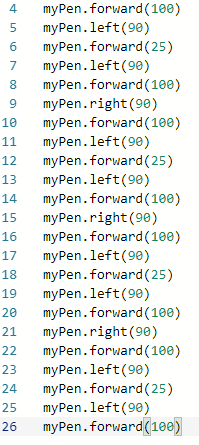
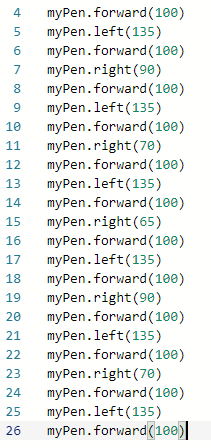
a. Provide a listing of your program code below:



5. Create a program to draw any three of the shapes described in “Part III” of

the PythonWorksheetII document.

a. Provide a listing of your program code below:



**Level 2: Using a Loop**

1. Google the keywords “Python Turtle Methods”.

a. Explain how the “goto” method works and how you could use it when drawing repeated shapes.

The goto method works when you would want the turtle to return to a specific xand y coordinate.

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

They didn’t list how to repeat blocks of code to complete a more complex shape.

2. Create a repeating pattern on your screen. The pattern must meet the following requirements:

a. The basic pattern must be made up of several individual Turtle methods (e.g. changes of colour, changes of direction, size, motion, etc.)

b. The basic pattern must be repeated several times with a shift in starting position each time.

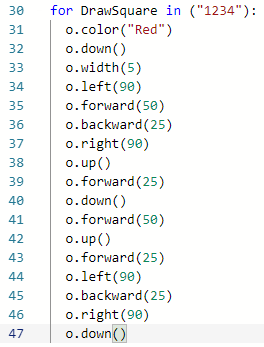
****

3. Use a Python Loop to create your repeating pattern

a. The Loop may be a Counted Loop or a Conditional Loop

b. The indented block of code for the loop should be your basic pattern.

4. Provide a listing of your repeating pattern loop below.



**Level 3: Defining a Function**

1. Google the keywords “Python Function Syntax”.

a. Explain what the “def” keyword does

Defines a function.

b. Explain any special rules regarding the function name

They must start with a letter or underscore, they can have numbers and capital letters, can’t be another operator or action, and can be any length.

c. Explain what the parameters (or arguments) do

They tell python the amount of the action you would want to perform. For example, if you would want to turn the python right, (90) would turn python 90 degrees., same thing for forward and backward, except that it’s distance instead of degrees because you’re not turning.

d. Where should the colon “:” be placed

This should be placed to tell python that there is more code to follow, and not just end that specific block of code there. For example, in the if statement, there are colons so that python looks at the next block of code to perform if the if statement is true.

e. Explain how to write Python statements that make up the function body

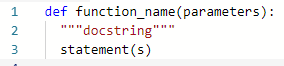
You would write the code how you would normally write it, but starting it after the : of the function name, and under it with an indentation to let python know that the code is what follows after the “:”.

f. Explain the “return” statement

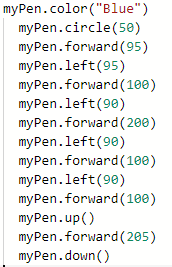
The return statement terminates the function, and then returns the code back to the main code that needs to be executed.

2. Provide an example of a simple function that uses one or more parameters.

a. Write the function definition below



b. Write some code to call the function below

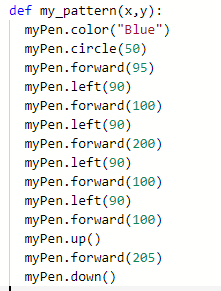


3. Convert your basic pattern (from Level 2 above) into a function

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



5. Use a your basic pattern function and a Python Loop to create your repeating pattern

a. The Loop may be a Counted Loop or a Conditional Loop

b. Your function should be called from within the loop.

6. Provide a listing of your function definition and repeating pattern loop below.

