**Outline**

t.b.d.

**Objectives**

* tbd

**Materials**

* tbd

**Level 0: Teacher Demo of Sample Programs**

1. Sample program #1 is an example of a "Syntax Error". Follow the teacher demo and explain the characteristics of a syntax error. Consider the following criteria:  
   1. Did the program have an error before starting to run?
   2. Did the program encounter an error before it finished running?
   3. Did the program do what it was supposed to do?

1. Sample program #2 is an example of a "Run-time Error". Follow the teacher demo and explain the characteristics of a run-time error. Consider the following criteria:  
   1. Did the program have an error before starting to run?
   2. Did the program encounter an error before it finished running?
   3. Did the program do what it was supposed to do?

1. Sample program #3 is an example of a "Logic Error". Follow the teacher demo and explain the characteristics of a logic error. Consider the following criteria:  
   1. Did the program have an error before starting to run?
   2. Did the program encounter an error before it finished running?
   3. Did the program do what it was supposed to do?

**Level 1: Syntax Errors**

1. Research the definition of the word "Syntax". Summarize its meaning below and how it relates to computer languages and programming.

**The**[**syntax**](https://en.wikipedia.org/wiki/Syntax_(programming_languages))**of the**[**Python programming language**](https://en.wikipedia.org/wiki/Python_(programming_language))**is the set of rules that defines how a Python program will be written and interpreted**

1. Research the definition of a "Syntax Error" related to computer programming. Summarize this definition below.

**Syntax errors – usually the easiest to spot, syntax errors occur when you make a typo.**

1. Explain why Sample Program #1 is an example of a "Syntax Error".

**Since the program did not start to run, it is syntax error.**

1. Find and correct the syntax errors in Sample Program #1. Provide a listing of your corrected program below.
   * Use a "#" at the beginning of each line containing an error   
     to "Comment Out" the bad code
   * List the corrected code line underneath the commented out error line

import turtle

myPen = turtle.Turtle()

circleColors = [(196,196,0),(196,0,196),(0,196,196)]

def drawCircle(rgb) :

myPen.down(

myPen.color(rgb)

myPen.begin\_fill()

myPen.circle(8)

myPen.end\_fill()

myPen.up()

myPen.forward(22)

circleNumber = 0

for circleIndex in range(3) :

**#** drawCircle(circleColors[circleNumber])

circleNumber = circleNumber + 1

**Level 2: Run-time Errors**

1. Research the definition of a "Run-time Error" related to computer programming. Summarize this definition below.

**A runtime error is a program error that occurs while the program is running.**

1. Explain why Sample Program #2 is an example of a "Run-time Error".

**Running time error happened when the program is running.**

1. Find and correct the run-time errors in Sample Program #2. Provide a listing of your corrected program below.
   * Use a "#" at the beginning of each line containing an error   
     to "Comment Out" the bad code
   * List the corrected code line underneath the commented out error line

import turtle

myPen = turtle.Turtle()

circleColours = [(196,196,0),(196,0,196),(0,196,196)]

def drawCircle(rgb) :

myPen.down()

myPen.color(rgb)

myPen.begin\_fill()

myPen.circle(8)

myPen.end\_fill()

myPen.up()

myPen.forward(22)

**#** circleNumber = 0

for circleIndex in range(4) :

drawCircle(circleColours[circleNumber])

circleNumber = circleNumber + 1

1. Explain the difference between a "syntax error" and a "run-time error".

**The difference is that syntax error happens before the program starts, and running time error is an error that happens when the program is running.**

**Level 3: Logic Errors**

1. Research the definition of a "Logic Error" related to computer programming. Summarize this definition below.

**Logical errors are the most difficult to fix. They occur when the program runs without crashing, but produces an incorrect result.**

1. Explain why Sample Program #3 is an example of a "Logic Error".

**It only printed two circles that are black.**

1. Find and correct the logic errors in Sample Program #3. Provide a listing of your corrected program below.
   * Use a "#" at the beginning of each line containing an error   
     to "Comment Out" the bad code
   * List the corrected code line underneath the commented out error line

import turtle

myPen = turtle.Turtle()

circleColours = [(196,196,0),(196,0,196),(0,196,196)]

def drawCircle(rgb) :

myPen.down()

**#** myPen.color(rgb)

myPen.begin\_fill()

myPen.circle(8)

myPen.end\_fill()

myPen.up()

myPen.forward(22)

numOfCircles = 3

**#** for circleIndex in range(3) :

circleNumber = numOfCircles - circleIndex - 1

drawCircle(circleColours[circleNumber])

1. Explain the difference between a "logic error" and a "syntax error".

**The difference is that syntax error happens before the program starts, and logic error occur when the program runs without crashing, but produces an incorrect result.**

1. Explain the difference between a "logic error" and a "run-time error".

**The difference is that logic error occur when the program runs without crashing, but produces an incorrect result, and running time error is an error that happens when the program is running.**

**Level 4: Your Sample Program**

1. Create a sample program to show the different types of programming errors. Provide your program listing below.
   * Your program must be of your own design and must be different from the sample programs provided in this module.
   * Your program must contain at least one example of each of: a syntax error, a run-time error, and a logic error.
   * Provide the corrected code in a comment underneath the error code (using a "#" at the beginning of the comment line).

import turtle

myPen = turtle.Turtle()

# These variables track the position of the turtle pen

posX = 0

posY = 0

# These variables define the image information.

# Each pixel in the image has a (r,g,b) value

# The complete image is simply a list of pixels

pixelAddress = 0

pixelMemory = [

# pixelMemory = [

(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(10,10,10),(10,10,10),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),

(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(10,10,10),(250,250,0),(250,250,0),(10,10,10),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),

(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(10,10,10),(250,250,0),(250,250,0),(10,10,10),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),

(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(10,10,10),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(10,10,10),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),

i

# (10,10,10),(10,10,10),(10,10,10),(10,10,10),(10,10,10),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(10,10,10),(10,10,10),(10,10,10),(10,10,10),(10,10,10),(1000000,100000,1000000),

(1000000,100000,1000000),

(10,10,10),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(10,10,10),(250,250,0),(10,10,10),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(10,10,10),(1000000,100000,1000000),

(1000000,100000,1000000),(1000000,100000,1000000),(10,10,10),(250,250,0),(250,250,0),(250,250,0),(10,10,10),(250,250,0),(10,10,10),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(10,10,10),(1000000,100000,1000000),

(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(10,10,10),(250,250,0),(250,250,0),(250,250,0),(10,10,10),(250,250,0),(10,10,10),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(10,10,10),(1000000,100000,1000000),

(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(10,10,10),(250,250,0),(250,250,0),(10,10,10),(250,250,0),(10,10,10),(250,250,0),(250,250,0),(250,250,0),(10,10,10),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),

(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(10,10,10),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(10,10,10),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),

(1000000,100000,1000000),(1000000,100000,1000000),(10,10,10),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(10,10,10),(1000000,100000,1000000),(1000000,100000,1000000),

(1000000,100000,1000000),(1000000,100000,1000000),(10,10,10),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(10,10,10),(1000000,100000,1000000),(1000000,100000,1000000),

(1000000,100000,1000000),(10,10,10),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(10,10,10),(10,10,10),(10,10,10),(10,10,10),(250,250,0),(250,250,0),(250,250,0),(250,250,0),(10,10,10),(1000000,100000,1000000),(1000000,100000,1000000),

(10,10,10),(250,250,0),(250,250,0),(10,10,10),(10,10,10),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(10,10,10),(10,10,10),(250,250,0),(250,250,0),(10,10,10),(1000000,100000,1000000),

(1000000,100000,1000000),(10,10,10),(10,10,10),(10,10,10),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(10,10,10),(10,10,10),(10,10,10),

(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),(1000000,100000,1000000),

]

# This user defined function draws a single image pixel

def drawPixel(rgb) :

global posX

myPen.down()

myPen.color(rgb)

myPen.begin\_fill()

myPen.circle(8)

myPen.end\_fill()

myPen.up()

myPen.forward(18)

posX = posX + 18

# This user defined function starts a new row of pixels

def newRow() :

global posX

global posY

myPen.up()

myPen.left(180)

myPen.forward(posX)

myPen.left(90)

myPen.forward(18)

myPen.left(90)

myPen.down()

posX = 0

posY = posY + 18

# THE MAIN PROGRAM CODE STARTS HERE

#

# Draw eight rows of the image.

# Each row contains eight pixels

for row in range (16) :

for column in range(6) :

# for column in range(6) :

drawPixel(pixelMemory[pixelAddress])

pixelAddress += 1

newRow()

**SAMPLE PROGRAM #1 - Syntax Error**

import turtle

myPen = turtle.Turtle()

circleColors = [(196,196,0),(196,0,196),(0,196,196)]

def drawCircle(rgb) :

myPen.down(

myPen.color(rgb)

myPen.begin\_fill()

myPen.circle(8)

myPen.end\_fill()

myPen.up()

myPen.forward(22)

circleNumber = 0

for circleIndex in range(3) :

drawCircle(circleColours[circleNumber])

circleNumber = circleNumber + 1

**SAMPLE PROGRAM #2 - Run-time Error**

import turtle

myPen = turtle.Turtle()

circleColours = [(196,196,0),(196,0,196),(0,196,196)]

def drawCircle(rgb) :

myPen.down()

myPen.color(rgb)

myPen.begin\_fill()

myPen.circle(8)

myPen.end\_fill()

myPen.up()

myPen.forward(22)

circleNumber = 1

for circleIndex in range(4) :

drawCircle(circleColours[circleNumber])

circleNumber = circleNumber + 1

**SAMPLE PROGRAM #3 - Logic Error**

import turtle

myPen = turtle.Turtle()

circleColours = [(196,196,0),(196,0,196),(0,196,196)]

def drawCircle(rgb) :

myPen.down()

myPen.begin\_fill()

myPen.circle(8)

myPen.end\_fill()

myPen.up()

myPen.forward(22)

numOfCircles = 3

for circleIndex in range(2) :

circleNumber = numOfCircles - circleIndex - 1

drawCircle(circleColours[circleNumber])