**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?  
      Yes it had one before it started to run, the code was not right.
   2. Did the program encounter an error before it finished running?  
      This does not occur as it had an error before it started to run.
   3. Did the program do what it was supposed to do?

This also does not apply

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?  
      No it did not.
   2. Did the program encounter an error before it finished running?  
      Yes sample two had a runtime error meaning it was running but there was an error in the code Python could not read and stopped as soon as that point took place.
   3. Did the program do what it was supposed to do?

No as there was a run time error it did not fully complete so it did not do its job.

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?  
      No it did not, it ran successfully.
   2. Did the program encounter an error before it finished running?  
      Yes it did as there was a logic error Python could not understand or it did not make sense in general.
   3. Did the program do what it was supposed to do?

No it did not, it didn’t complete.

**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.

Syntax of the Python programming language is the set of rules that defines how the program is written as well as how it interpreted the sentence.

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

A syntax error in computer programming means there is a line that python does not know what to do with. Therefore, it considers it as a syntax error.

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

It is because line 4 and 7 is a wrong string and python turtle cannot process it in their systems.

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
2. import turtle
3. myPen = turtle.Turtle()
4. #circleColors = [
5. circleColours
6. ( 196,196,0),(196,0,196),(0,196,196)
7. ]
8. def drawCircle(rgb) :
9. #myPen.down(
10. myPen.down()
11. myPen.color(rgb)
12. myPen.begin\_fill()
13. myPen.circle(8)
14. myPen.end\_fill()
15. myPen.up()
16. myPen.forward(22)
17. circleNumber = 0
18. for circleIndex in range(3) :
19. drawCircle(circleColours[circleNumber])
20. circleNumber = circleNumber + 1

In order to fix the errors on line 4 since we are Canadians circleColors should be circleColours and on line 7 myPen.down( should have the extra bracket and be myPen.down()

**Level 2: Run-time Errors**

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

A run time error is a program error which will occur when the program is running. The program would start and stop out of nowhere

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

It is an example of a run time error because line 18’s list is out of range and it is easily fixable as well as the circle number and one thing has to be taken out.

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
2. import turtle
3. myPen = turtle.Turtle()
4. circleColours = [(196,196,0),(196,0,196),(0,196,196)]
5. def drawCircle(rgb) :
6. myPen.down()
7. myPen.color(rgb)
8. myPen.begin\_fill()
9. myPen.circle(8)
10. myPen.end\_fill()
11. myPen.up()
12. myPen.forward(22)
13. #circleNumber = 1
14. CircleNumber=0
15. #for circleIndex in range(4) :
16. for circleIndex in range(3) :
17. print(circleNumber)this should not be in the code
18. drawCircle(circleColours[circleNumber])
19. circleNumber = circleNumber + 1

List the corrected code line underneath the commented out error line

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

The difference between it is that a runtime error is a program error that occurs while the program is running. Syntax error is when the program cannot understand or process a certain line so it prevents the code from even starting.

**Level 3: Logic Errors**

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

They occur when the program runs without crashing but produces an incorrect result. It can be caused by a mistake in the program’s logic

1. Explain why Sample Program #3 is an example of a "Logic Error".
2. 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
3. import turtle
4. myPen = turtle.Turtle()
5. circleColours = [(196,196,0),(196,0,196),(0,196,196)]
6. def drawCircle(rgb) :
7. myPen.down()

myPen.colour(rgb) this is not in the program so it makes no black circles this will make them coloures

1. myPen.begin\_fill()
2. myPen.circle(8)
3. myPen.end\_fill()
4. myPen.up()
5. myPen.forward(22)
6. #numOfCircles = 2
7. numOfCircles = 3
8. for circleIndex in range(0) :
9. circleNumber = numOfCircles - circleIndex
10. print(circleNumber) This is not needed in the code
11. drawCircle(circleColours[circleNumber])

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

The difference is that it happens within the program and not in the code. Syntax error is a problem with the string or code while logic is pointing towards the logical sense of the program.

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

Run time error happens when there is a problem midway while logic is once again questioning the programs logic. Logic errors cannot be detected in the program as it is not a run time error or a syntax error

**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).   
  
 myPen.color(rgb)

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 = [

(218,234,94),(235,90),(154,151,61),(148,118,79),(201,176,95),(161,142,85),(149,122,64),(142,116,73),(209,212,129),(229,243,128),(232,242,121),(222,248,66),(211,218,54),(202,137,52),(160,119,67),(66,37,29),(64,27,15),(166,113,62),(228,234,115),(217,238,107),(255,255,217),(249,251,183),(255,255,158),(207,177,116),(187,184,175),(153,107,51),(144,106,65),(117,91,62),(200,204,82),(232,250,120),(214,229,116),(233,245,148),(237,231,210),(160,147,141),(180,146,91),(246,166,45),(146,84,16),(54,29,20),(151,145,88),(249,255,147),(198,216,99),(215,239,95),(216,217,194),(148,144,157),(238,197,126),(220,152,53),(132,71,4),(75,50,32),(151,146,132),(237,245,149),(220,230,130),(220,236,108),(242,233,208),(214,205,205),(214,203,185),(197,179,168),(147,119,82),(159,146,113),(181,163,128),(145,103,38),(221,243,113),(201,190,97),(221,188,165),(239,228,209),(166,150,127),(186,183,172),(130,122,100),(108,88,50),(102,61,18),(122,59,8),(229,241,126),(163,135,93),(202,183,165),(238,225,190),(151,126,82),(125,97,53),(92,55,14),(68,19,0),(82,32,0),(123,72,16)

]

# This user defined function draws a single image pixel

def drawPixel(rgb) :

global posX

myPen.down()

# logic error there is no rgb value to make colour

myPen.color(rgb)

myPen.begin\_fill()

myPen.circle(8)

myPen.end\_fill()

myPen.up()

myPen.forward(0)

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. – run time error as it goes off screen it is hard to fix

posX = 18

posY = posY + 18

# THE MAIN PROGRAM CODE STARTS HERE

#

# Draw eight rows of the image.

# Each row contains eight pixels

for row in range (8) :

for column in range(10) :

drawPixel(pixelMemory[pixelAddress])

pixelAddress += 1

# newRow – Syntax error as the program cannot handle it

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) :

print(circleNumber)

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 = 2

for circleIndex in range(2) :

circleNumber = numOfCircles - circleIndex

print(circleNumber)

drawCircle(circleColours[circleNumber])