**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 did not do anything in turtle. The black screen had an error saying “bad input”.
   2. Did the program encounter an error before it finished running?  
      b does not apply, because the program did not run in the first place.
   3. Did the program do what it was supposed to do?  
      c does not apply because the program did not run in the first place.
2. 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 began to run.
   2. Did the program encounter an error before it finished running?  
      Yes, it only completed two out of three circles, and displayed an error stating “list index out of range on line 17”.
   3. Did the program do what it was supposed to do?

No it did not do what it was supposed to because it did not finish printing all of the circles.

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 began to run.
   2. Did the program encounter an error before it finished running?  
      No it did not encounter any python error, it just stopped running and only completed 2 circles.
   3. Did the program do what it was supposed to do?

The program did not do what it was supposed to because it only completed 2 out of 3 circles which were in black when they were supposed to be in different colours.

**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 word “Syntax” means to have a well-formed arrangement of words and phrases in a certain language. In computers a “Syntax” means the spelling and grammar of a programing language. It refers to the correct use of words and phrases in programing. “Syntax” can also refer to a set of rules.

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

The definition of a “Syntax Error” in regards to computer programming is when a string or character is incorrectly placed in a command, and due to the inaccuracy the command fails in execution.

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

Sample Program #1 is an example of a “Syntax Error” because an error was made in the code, which led to the failure of execution.

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. #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 = 0
14. for circleIndex in range(3) :
15. drawCircle(circleColours[circleNumber])
16. 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 run-time error occurs which the program is running. The term is often used in contrast to other types of program errors, such as syntax errors and compile time errors. There are many different types of runtime errors.

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

Sample Program #2 is an example of a Run-time error because the program started to run however, it stopped half way and did not complete all the circles, due to the in correct number in line 14 for ‘circlenumber’. A run-time error stopped while the program is running due to an error in the program and that was what happened in this program.

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
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 = 0
14. for circleIndex in range(4) :
15. drawCircle(circleColours[circleNumber])
16. circleNumber = circleNumber + 1

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

The difference between a “syntax error” and a “run-time error” is that when writing the code if a syntax error was made the whole code will not work, however a run-time error lets the program run until it get to the error.

**Level 3: Logic Errors**

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

A logic error produces unintended or undesired output or other behavior, although it may not immediately be recognized as such.

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

Sample Program #3 is an example of a “Logic Error” because there was no actual error that python detected, however the program still did not do what it was supposed to, making it a Logic Error. The program printed out black circles instead of coloured ones because a line of code was missing “myPen.color(rgb)”.

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
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. numOfCircles = 3
14. #for circleIndex in range(3) :
15. circleNumber = numOfCircles - circleIndex - 1
16. drawCircle(circleColours[circleNumber])

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

A logic error goes through with the program and does not detect an error because python can read the code without a line being there and works with what is already there. However, if there is a syntax error (a language mistake in the code), the program does not run at all.

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

The difference between a logic error and run-time error is that the run- time error runs the program but stops when it detects an error. But a logic error finishes the code because it does not read any mistakes even if it may not be what the person wants it to do.

**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).

**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])