**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?

There was an error before it started to run

* 1. Did the program encounter an error before it finished running?

The program didn’t start to run so there was no error before the program finished running

* 1. Did the program do what it was supposed to do?  
     No because the program didn’t start to run

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, the program ran

* 1. Did the program encounter an error before it finished running?

Yes, it encountered an error before it finished

* 1. Did the program do what it was supposed to do?  
     No, the program was supposed to draw 3 circles but it only drew two

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?

The program started to run with no errors

* 1. Did the program encounter an error before it finished running?

No, the program didn’t encounter any errors, it just stopped running

* 1. Did the program do what it was supposed to do?

The program didn’t do what it was supposed to do because there was a logical mistake in the written coding which caused the program to just stop running with no visible error.

**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 the arrangement of words and phrases to create well-formed sentences in a language. This relates to computer languages and programming because programs need to be able to read the codes that are being inputted into the computer. In this case, if the words and phrases are arranged in well-formed sentences, then the program will be able to read it and output it. Vice versa, if the words and phrases are not arranged into well-formed sentences, then there will be errors in the code and the program wouldn’t be able to pick it up.

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

In computer programming, a syntax error is an error in the sequence of words, characters, phrases that is intended to be written in a particular programming language.

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

Sample Program 1 is an example of a Syntax Error because the code is not written properly and so when the program is reading it, it can’t pick it up and won’t be able to output the information being inputted.

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
2. # myPen.begin\_fill()
3. # myPen.circle(8)
   * List the corrected code line underneath the commented out error line
4. import turtle
5. myPen = turtle.Turtle()
6. myPen.begin\_fill()
7. circleColors = [(196,196,0),(196,0,196),(0,196,196)]
8. def drawCircle(rgb) :
9. myPen.down()
10. myPen.color(rgb)
11. myPen.end\_fill()
12. myPen.up()
13. myPen.forward(22)
14. circleNumber = 0
15. for circleIndex in range(3) :
16. drawCircle(circleColours[circleNumber])
17. circleNumber = circleNumber +

**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. This term is often used in contrast to other types of program errors, such as syntax.

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 list index is out of range on one of the lines. In addition, there is an index and indexes start like 0, 1 ,2 ,3, etc. But when the debugging code was written, it wrote 1, 2, 3 but there’s no third index that is defined. Within the square brackets, there are only 0, 1, 2.

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

# drawCircle(circleColours[circleNumber]

* + 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 = 1

for circleIndex in range(4) :

print(circleNumber)

circleNumber = circleNumber + 1

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

The difference between a syntax error is that there is a mistake within the coding that makes it not able to run. A run-time error can have correct coding but it can stop running if the program has a logic error.

**Level 3: Logic Errors**

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

A logic error in computer programming is a bug in a program that causes it to operate incorrectly but not stop the entire program from running. A logic error produces unintended or undesired output or other behavior.

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 is a written code in the program that doesn’t make logical sense in the sense of computer programming. The program ran with no errors but it just stopped because there was a logic mistake.

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

# for circleIndex in range(2) :

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.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".

Syntax errors occur when a program does not conform to the grammar of a program of a programming language and it won’t be able to understand the program code. Logic errors occur when a program does not do what the programmer expects it to do.

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

A runtime error is a program error that occurs while a program is running. A logic error often lead to runtime errors but a logical error occurs when there is a logical error in the program, when there is something wrong in the algorithm.

**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).

**Logic Error**

1. import turtle
2. myPen=turtle.Turtle()
3. myPen.begin\_fill()
4. myPen.left(90)
5. myPen.forward(100)
6. myPen.right(135)
7. myPen.forward(140)
8. myPen.right(135)
9. myPen.forward(100)
10. myPen.end\_fill()
11. myPen.up()
12. myPen.right(90)
13. myPen.down()
14. myPen.forward(100)
15. myPen.right(90)
16. myPen.forward(100)
17. myPen.right(90)
18. myPen.forward(100)
19. numOfTriangles = 2
20. #numfTriangles=2 error

**Syntax Error**

import turtle

myPen=turtle.Turtle()

myPen.begin\_fill()

myPen.left(90)

myPen.forward(100)

myPen.right(135)

myPen.forward(140)

myPen.right(135)

myPen.forward(100)

myPen.end\_fill()

#myPen.u()

myPen.right(90)

myPen.down()

myPen.forward(100)

myPen.right(90)

myPen.forward(100)

myPen.right(90)

myPen.forward(100)

numfTriangles=2

**Run-Time Error**

import turtle

myPen=turtle.Turtle()

myPen.begin\_fill()

myPen.left(90)

myPen.forward(100)

myPen.right(135)

myPen.forward(140)

myPen.right(135)

myPen.forward(100)

myPen.end\_fill()

myPen.up()

#myPen.u()

myPen.right(90)

myPen.down()

myPen.forward(100)

myPen.right(90)

myPen.forward(100)

myPen.right(90)

myPen.forward(100)

numfTriangles=2

**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])