**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. When referring to a programming language, syntax is a set of rules for grammar and spelling. In other words, it means using character structures that a computer can interpret. For example, if a user tries to execute a command without proper syntax, it generates syntax error, usually causing the program to fail**

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

* **In computer science, a syntax error is an error in the syntax of a sequence of characters or tokens that is intended to be written in a particular programming language. For compiled languages, syntax errors are detected at compile-time. A program will not compile until all syntax errors are corrected.**

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

* **Sample Program #1 is an example of a Syntax Error because it does not start right away and there is a problem with the spelling/programming language.**

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() #missing other bracket

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]) #circlecolours is spelled wrong, renamed into "circlecolors" which has the assignment of the colors of the circles

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. 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. One example is a logic error, which produces the wrong output**

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 error occurred while the program was running but it stopped in the middle of the 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

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 #made into 0 because the index starts from 0

for circleIndex in range(3) : #made it into 3 because the index has only 3 values

drawCircle(circleColours[circleNumber])

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 a syntax error does not let the program run right away and the code has a spelling/programming language mistake. While a run-time error starts up the program but stops before completing the error which results in the program not doing what it was supposed to do. This means that there are problems with the code.**

**Level 3: Logic Errors**

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

**A logic error (or logical error) is a mistake in a program's source code that results in incorrect or unexpected behavior. It is a type of runtime error that may simply produce the wrong output or may cause a program to crash while running. ... Even small typos that do not produce syntax errors may cause logic errors**

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

Sample Program #3 is an example of a Logic error because although the program runs and python does not recognize any problem with the code, it still does not provide the proper results .

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) #missing code for color of circles

  myPen.begin\_fill()

  myPen.circle(8)

  myPen.end\_fill()

  myPen.up()

  myPen.forward(22)

numCircles = 3

for circleIndex in range(3) : #changed to 3 because there are 3 circles

  circleNumber = numCircles - circleIndex - 1

  drawCircle(circleColours[circleNumber])

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

**The difference between syntax errors and logic errors. Syntax errors occur when a program does not conform to the grammar of a programming language, and the compiler cannot compile the source file. 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 run-time error starts the program but then stops it before it can finish. This means while the program was running, it found an error in the code which prevented the program from completing.**

**A logical error is simply that the programmer is doing something wrong in their code which makes the program come out as not expected. Python does not recognize it because there is no problem with the code but the program does not result in what the user wants it to result in**

**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()

or i in range(4): # Syntax error, Fix: for i in range(4):

myPen.right(90)

myPen.forward(100)

myPen.left(90)

myPen.forward(50)

myPen.left(90)

myPen.forward(100)

myPen.up()

myPen.foward(250) # Run-time error, Fix; myPen.forward(250)

myPen.right(90)

myPen.down()

for x in range(3):

myPen.forward(100)

myPen.left(90)

myPen.forward(100)

myPen.up()

myPen.forward(600)

myPen.down()

myPen.color("blue")

myPen.circle(50)

myPen.color("re") #Logic Error, square supposed to turn out red, Fix: myPen.color("red")

myPen.forward(50)

myPen.left(90)

for h in range(4):

myPen.forward(100)

myPen.left(90)