**Questions**

**Basics**

1. What is an expression?
   1. A combination of values, variables, operations, and functions called
2. What is a syntax error?
   1. A syntax error is what happens when a compiler can’t understand source code when making machine code
3. What is PEP8?
   1. A document that provides guidelines on how to write Python
4. What does a linter do?
   1. A linter is a static code analysis tool that makes sure programming errors and bugs don’t occur
5. What is the result of this expression: “\*” \* 10
   1. It prints the string “\*” 10 times
6. What is CPython?
   1. CPython is a reference implementation for Python made from a mixture of C and Python.
7. How is CPython different from Jython?
   1. Jython is CPython but using Java instead of C
8. How is CPython different from IronPython?
   1. IronPython is CPython but using C# instead of C

**Primitive Types**

1. What is a variable?
   1. Variables are a reference to a position in memory. The value can be changed
2. What are the primitive built-in types in Python?
   1. Integer, Float, Boolean, String
3. When should we use “”” (tripe quotes) to define strings?
   1. Triple quotes in python ignore special characters like line break slashes, single quotes, etc
4. Assuming (name = “John Smith”), what does name[1] return?
   1. It returns “o” since that’s the character at position 1
5. What about name[-2]?
   1. It returns “t”, since that’s the second last character
6. What about name[1:-1]?
   1. It returns a substring from index 1 to the second last character
7. How to get the length of name?
   1. Len(name)
8. What are the escape sequences in Python?
   1. Backslash, single quotes, double quotes, newline, and backspace
9. What is the result of f**“{2+2}+{10%3}”**?
   1. 4+1
10. Given (name = “john smith”), what will **name.title()** return?
    1. John Smith (.title capitalizes the first character of each word and lowers all the rest)
11. What does name.**strip()** do?
    1. .strip removes the whitespaces in a string
12. What will **name.find(“Smith”)** return?
    1. It returns the index of the string “Smith”, which here is -1 because its case sensitive
13. What will be the value of **name** after we call name.**replace(“j”, “k”)**?
    1. kohn smith
14. How can we check to see if **name** contains **“John”**?
    1. Print(“John” in name);
15. What are the 3 types of numbers in Python?
    1. Int, float, complex

**Control Flow**

* What is the difference between **10 / 3** and **10 // 3**?
  1. 10/3 = 3.33333, 10//3 = 3. / is float division, // is integer division
* What is the result of 10 \*\* 3?
  1. Exponential (10^3), so 10 \*\* 3 = 1000
* Given (**x = 1**), what will be the value of after we run (**x +=2**)?
  1. X=3
* How can we round a number?
  1. Use the .round() function
* What is the result of **float(1)**?
  1. Converts the integer to a float, so 1.0
* What is the result of **bool(“False”)**?
  1. Returns true, because “False” is just a string in this case, and bool() just checks for content
* What are the false values in Python?
  1. False values are anything in bool() that will make it return false, which are
  2. False
  3. None
  4. 0
  5. “”
  6. ()
  7. []
  8. {}
* What is the result of **10 == “10”**?
  1. False due to type mismatch
* What is the result of **“bag” > “apple”**?
  1. True because it compares the first character of each string, and b is a “bigger” character because it’s later in the alphabet
* What is the result of **not(True or False)**?
  1. True or False will always return true, so not of that is false
* Under what circumstances does the expression **18 <= age < 65** evaluate to True**?** 
  1. If age is anywhere from 18 to 64. NOT 65
* What does **range(1, 10, 2)** return?
  1. 1,3,5,7,9
* Name 3 iterable objects in Python.
  1. List, tuples, dictionaries, sets

**Functions**

1. What is the difference between a parameter and an argument?
   1. Parameters are listed in the functions definition, arguments are values passed to the function. In other words, parameters are initialized to the argument values
2. All functions in Python by default return …?
   1. A default value
3. What are keyword arguments and when should we use them?
   1. Keyword arguments are arguments that get assigned to their parameter using “=”. These are used to make function calls more explicit
4. How can we make a parameter of a function optional?
   1. A parameter can be optional by assigning it a default value so it’s still filled but doesn’t have to be defined every time a user calls the function
5. What happens when we prefix a parameter with an asterisk (\*)?
   1. We use it sometimes when passing iterables. It allows us to pass all items in a specific iterable to another function
6. What about two asterisks (\*\*)?
   1. It does the same thing but with keyword arguments
7. What is scope?
   1. The range of a program where a variable/constant/function is recognized
8. What is the difference between local and global variables?
   1. Local variables are declared inside functions and are only accessible in the one, global variables are declared outside a function and can be accessed anywhere at any time
9. Why is using the **global** statement a bad practice?
   1. Because it can be accessed from multiple places at once and changed simultaneously, resulting in inconsistent values

**Coding Exercises**

1. Write a function that returns the maximum of two numbers.

def maximum(a, b)

if a > b:

print(f”{a} is the maximum”)

else:

print(f”{b} is the maximum”)

2. Write a function called **fizz\_buzz** that takes a number.

1. If the number is divisible by 3, it should return “Fizz”.

2. If it is divisible by 5, it should return “Buzz”.

3. If it is divisible by both 3 and 5, it should return “FizzBuzz”.

4. Otherwise, it should return the same number.

def fizz\_buzz(num):

if(num % 5 == 0 && num % 3 == 0):

return “FizzBuzz”

elif(num % 3 == 0):

return “Fizz”

elif(num % 5 == 0):

return “Buzz”

else:

return num

3. Write a function for checking the speed of drivers. This function should have one parameter: speed.

1. If speed is less than 70, it should print “Ok”.

2. Otherwise, for every 5km above the speed limit (70), it should give the driver one demerit point and print the total number of demerit points. For example, if the speed is 80, it should print: “Points: 2”.

3. If the driver gets more than 12 points, the function should print: “License suspended”

def speedCheck(speed)

points = 0

if(speed < 70):

print(“OK”)

else:

points = int((speed – 70)/5)

print(f”Points: {points}”)

if points >= 12:

print(“License suspended”)

4. Write a function called **showNumbers** that takes a parameter called **limit.** It should print all the numbers between 0 and limit with a label to identify the even and odd numbers. For example, if the limit is 3, it should print:

o 0 EVEN

o 1 ODD

o 2 EVEN

o 3 ODD

def showNumbers(limit):

for count in range(limit+1):

if count % 2 == 0:

print(f”{count} EVEN”)

else:

print(f”{count} ODD”)

5. Write a function that returns the sum of multiples of 3 and 5 between 0 and **limit** (parameter). For example, if limit is 20, it should return the sum of 3, 5, 6, 9, 10, 12, 15, 18, 20.

def threeAndFive(limit):

sum = 0

for count in range(limit+1):

if count % 3 == 0 or count % 5 == 0:

sum += count

print(sum)

6. Write a function called **show\_stars(rows).** If **rows** is 5, it should print the following:

* \*
* \*\*
* \*\*\*
* \*\*\*\*
* \*\*\*\*\*

def show\_stars(rows):

for count in range(1, rows+1):

print(“\*” \* count)

7. Write a function that prints all the prime numbers between 0 and **limit** where limit is a parameter.

def primeList(limit):

for num in range(1, limit+1):

if num > 1:

for i in range(2, num):

if (num % i) == 0:

break

else:

print(num)