CS118
Library Functions

For each exercise, provide a separate .PY file that performs the specified task.
For each of the programs provided, use the command: `print('\x1b[2J')` to clear the console first.
Italicized questions should be answered in the code as a comment.

Trig Functions

1. Write a Python3 program that imports the `math` module, then stores in the variable `angle_d` a hardcoded angle in degrees (0-360°). Using the `angle_d` variable, have the program use the `math.radians()` function convert the value in `angle_d` to radians, storing the result in the variable `angle_r`. Use the `math.sin()` functions to compute the sine of the variable `angle_d` and store result in `sin_d`; then use `math.sin()` to compute the sine of the variable `angle_r` and store the result in the variable `sin_r`. Use the `print()` function with the variables to show each: the angle in degrees and radians; the sine of the angle using the `sin()` function with `angle_d`; and the sine of the angle using the `sin()` function with `angle_r`. *Which computed value of the sine is correct? What expectation of the function results in the other being incorrect?*

```
# Clear the console
print("\x1b[2J")

import math
angle_d = 90
angle_r = math.radians(angle_d)
sin_d = math.sin(angle_d)
sin_r = math.sin(angle_r)
print(angle_d)
print(angle_r)
print(sin_d)
print(sin_r)

# The value in sin_r is correct.
# The math.sin() function assumes the angle provided will be in radians
```

2. Write a MATLAB program that has a hardcoded angle in degrees (0-360°). Use the `sin()` and `cos()` functions to compute the tangent of the angle, assuming the angle will never be an odd multiple of 90°. Do not use the `tan()` function – apply your knowledge of trig identities. Show the angle, sine, cosine, and tangent of the angle by use of `print()`.

```
# Clear the console
print("\x1b[2J")

import math
angle_d = 45
angle_r = math.radians(angle_d)

sin_r = math.sin(angle_r)
cos_r = math.cos(angle_r)
tan_r = sin_r / cos_r

print(angle_r)
print(sin_r)
print(cos_r)
print(tan_r)
```

Logarithms

3. The population for an exponentially-growing society follows this mathematical formula:

$$y = y_0 e^{kt}$$

where y_0 is the initial population, k is the growth constant, and y is the population after time t.

Make a Python3 program that uses the `math.exp()` and `math.floor()` functions to compute the population of bacteria after t minutes elapses, where t is a hardcoded value. The initial population is 10 bacteria. Hardcode the growth constant as 0.1234. Display the initial and final population along with the time period by use of `print()`.

```
# Clear the console
print("\x1b[2J")

import math
t = 33
y0 = 10
k = 0.1234
y = math.floor(y0 * math.exp(k*t))

print(y0)
print(y)
```

4. On paper, re-arrange the mathematical expression in #3 to solve for *k* – there is no need to submit the paper since you will then write a Python3 program that uses Python's `math.log()` function and the re-arranged math to compute *k,* assuming the population doubles every five minutes. Have the program display *k* by use of `print()`.

```
# Clear the console
print("\x1b[2J")

import math

# The Math:
# ln(y/y0) = k*t
# ln(2) = k*t
# k = ln(2)/t

t = 5
k = math.log(2) / t

print(k)
```

Random Numbers

5. Prepare a Python3 program that uses the `random` module's `uniform()` function three times to generate three pseudo-random numbers, storing each in its own variable. The first value will lie between 10 and 15; the second value between 100 and 200; and the third value between the previous two randomly-generated values. Display each generated value by use of `print()`.

```
# Clear the console
print("\x1b[2J")

import random
pr1 = random.uniform(10, 15)
pr2 = random.uniform(100, 200)
pr3 = random.uniform(pr1, pr2)

print(pr1)
print(pr2)
print(pr3)
```

6. Make a copy of #5 and modify it so that the values generated are integers, not floats.

```
# Clear the console
print("\x1b[2J")

import random
pr1 = random.randint(10, 15)
pr2 = random.randint(100, 200)
pr3 = random.randint(pr1, pr2)

print(pr1)
print(pr2)
print(pr3)
```

Files and Directories

7. Prepare a Python3 program that uses the `os` module's `getcwd()` function to obtain the complete path for the program, saving it in the variable `my_path`. Display `my_path` by use of `print()`.

```
# Clear the console
print("\x1b[2J")

import os

my_path = os.getcwd()
print(my_path)
```