PCC-DS 391 Lab Assignment 4

1. Write a program to do the following task.
    a. Takes a number and returns its square root using math.sqrt().
    b. Takes a number and returns the factorial of a given number using the math.factorial().
    c. Takes two numbers and returns the result of x raised to the power y using math.pow().
    d. Takes two numbers and returns the greatest common divisor of two numbers using math.gcd().
    e. Takes a floating point number and returns its ceiling and floor using math.ceil() and math.floor().

```
import math

def calculate_square_root(x):
    return math.sqrt(x)

def calculate_factorial(n):
    return math.factorial(n)

def calculate_power(x, y):
    return math.pow(x, y)

def calculate_gcd(x, y):
    return math.gcd(x, y)

def calculate_ceiling_and_floor(x):
    ceiling = math.ceil(x)
    floor = math.floor(x)
    return (ceiling, floor)

# Example usage:
# a. Calculate square root
number = 16
print(f"Square root of {number}: {calculate_square_root(number)}")

# b. Calculate factorial
number = 5
print(f"Factorial of {number}: {calculate_factorial(number)}")

# c. Calculate power
x = 2
y = 3
print(f"{x} raised to the power {y}: {calculate_power(x, y)}")

# d. Calculate greatest common divisor
a = 54
b = 24
print(f"Greatest common divisor of {a} and {b}: {calculate_gcd(a, b)}")

# e. Calculate ceiling and floor
```

```
floating_number = 4.7
ceiling, floor = calculate_ceiling_and_floor(floating_number)
print(f"Ceiling of {floating_number}: {ceiling}, Floor of {floating_number}: {floor}")
```

Output:

```
Square root of 16: 4.0
Factorial of 5: 120
2 raised to the power 3: 8.0
Greatest common divisor of 54 and 24: 6
Ceiling of 4.7: 5, Floor of 4.7: 4
```

2. Write a program that returns a random integer between a and b using random.randint() and returns a random floating-point number between 0 and 1 using random.random().

```
import random
def generate_random_integer(a, b):
   return random.randint(a, b)
def generate_random_float():
   return random.random()
# Example usage:
# Generate a random integer between a and b
a = 5
b = 15
random_integer = generate_random_integer(a, b)
print(f"Random integer between {a} and {b}: {random_integer}")
# Generate a random floating-point number between 0 and 1
random_float = generate_random_float()
print(f"Random floating-point number between 0 and 1: {random_float}")
```

Output:

```
Random integer between 5 and 15: 5
Random floating-point number between 0 and 1: 0.4700519042851178
```

3. Write a program that takes a list to do the following task
    a. Returns a randomly shuffled version of the list using random.shuffle().
    b. Returns a random element from a list using random.choice().

```
import random

def shuffle_list(lst):
   # Shuffle the list in place and return it
   random.shuffle(lst)
   return lst

def choose_random_element(lst):
   # Return a random element from the list
   return random.choice(lst)

# Example usage:
```

```
my_list = [1, 2, 3, 4, 5]

# a. Get a randomly shuffled version of the list
shuffled_list = shuffle_list(my_list)
print(f"Shuffled list: {shuffled_list}")

# b. Get a random element from the list
random_element = choose_random_element(my_list)
print(f"Random element from the list: {random_element}")
```

Output:

```
Shuffled list: [5, 2, 1, 3, 4]
Random element from the list: 2
```

4. Write a program that returns the mean (average), median, mode, variance, standard deviation of a list of numbers using python statistics module.

```
import statistics

numbers_list = [1, 2, 3, 4, 5, 5, 6, 7, 8]

mean = statistics.mean(numbers_list)
median = statistics.median(numbers_list)
mode = statistics.mode(numbers_list)
variance = statistics.variance(numbers_list)
stdev = statistics.stdev(numbers_list)

print(f"Mean: {mean}")
print(f"Median: {median}")
print(f"Mode: {mode}")
print(f"Variance: {variance}")
print(f"Standard Deviation: {stdev}")
```

Output:

```
Mean: 4.555555555555555
Median: 5
Mode: 5
Variance: 5.277777777777778
Standard Deviation: 2.2973414586817036
```

5. Write a program that takes a list of numbers and returns a new list with each number squared using the map() function and a lambda function.

```
def square_list(numbers):
    # Use map() with a lambda function to square each number in the list
    return list(map(lambda x: x**2, numbers))

# Example usage:
numbers_list = [1, 2, 3, 4, 5]
```

```
squared_numbers = square_list(numbers_list)
print(squared_numbers)  # Output: [1, 4, 9, 16, 25]
```

Output:

[1, 4, 9, 16, 25]

6. Write a program that filters out all even numbers from a list using the filter() function and a lambda function.

```
def filter_even(numbers):
    # Use filter() with a lambda function to keep only odd numbers
    return list(filter(lambda x: x % 2 != 0, numbers))
# Example usage:
numbers_list = [1, 2, 3, 4, 5, 6]
filtered_numbers = filter_even(numbers_list)
print(filtered_numbers)  # Output: [1, 3, 5]
```

Output:

[1, 3, 5]

7. Write a lambda function that takes two strings and concatenates them.

```
concatenate_strings = lambda str1, str2: str1 + str2
result = concatenate_strings("Hello, ", "world!")
print(result)
```

Output:

Hello, world!