ASSIGNMENT 8

Q1.

```python
class Person:
    def __init__(self, name, age, occupation):
        self._name = name
        self._age = age
        self._occupation = occupation

    # Getters
    def get_name(self):
        return self._name

    def get_age(self):
        return self._age

    def get_occupation(self):
        return self._occupation

    # Setters
    def set_name(self, name):
        self._name = name

    def set_age(self, age):
        self._age = age

    def set_occupation(self, occupation):
        self._occupation = occupation

```

Q2.

```python
from collections import Counter
from statistics import median

class Stats:
    def __init__(self, values):
        self._values = values

    def calculate_mean(self):
        return sum(self._values) / len(self._values) if self._values else None

    def calculate_median(self):
        return median(self._values) if self._values else None

    def calculate_mode(self):
        # Using Counter to find the mode(s)
        counter = Counter(self._values)
        modes = counter.most_common()
        max_count = modes[0][1]

        # Filter modes with maximum count
        modes = [mode[0] for mode in modes if mode[1] == max_count]

        return modes if modes else None

```

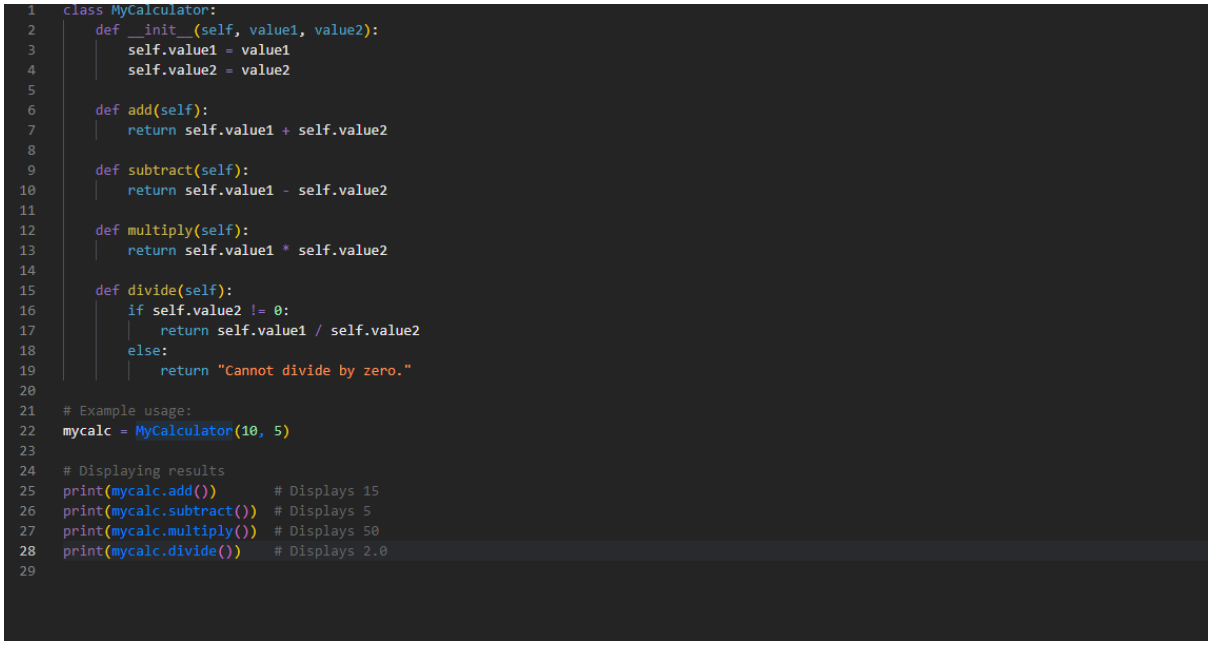
Q3.

```python
from abc import ABC, abstractmethod

class Shape(ABC):
    @abstractmethod
    def calculate_area(self):
        pass

class Triangle(Shape):
    def __init__(self, base, height):
        self.base = base
        self.height = height

    def calculate_area(self):
        return 0.5 * self.base * self.height

class Rectangle(Shape):
    def __init__(self, length, width):
        self.length = length
        self.width = width

    def calculate_area(self):
        return self.length * self.width

# Example usage:
triangle = Triangle(base=6, height=9)
rectangle = Rectangle(length=4, width=5)

print(f"Triangle Area: {triangle.calculate_area()}")  # Output: 27.0
print(f"Rectangle Area: {rectangle.calculate_area()}")  # Output: 20.0

```

Q4.

```python
class Vehicle:
    def __init__(self, brand, model, year):
        self.brand = brand
        self.model = model
        self.year = year

    def display_details(self):
        print(f"Brand: {self.brand}")
        print(f"Model: {self.model}")
        print(f"Year: {self.year}")

# Example usage:
car1 = Vehicle("Kia", "Morning", 2022)
car1.display_details()

car2 = Vehicle("Honda", "Accord", 2021)
car2.display_details()

```

Q5.



```python
class MyCalculator:
    def __init__(self, value1, value2):
        self.value1 = value1
        self.value2 = value2

    def add(self):
        return self.value1 + self.value2

    def subtract(self):
        return self.value1 - self.value2

    def multiply(self):
        return self.value1 * self.value2

    def divide(self):
        if self.value2 != 0:
            return self.value1 / self.value2
        else:
            return "Cannot divide by zero."

# Example usage:
mycalc = MyCalculator(10, 5)

# Displaying results
print(mycalc.add())       # Displays 15
print(mycalc.subtract())  # Displays 5
print(mycalc.multiply())  # Displays 50
print(mycalc.divide())    # Displays 2.0
```

THE ABOVE PICTURES ARE SCREENSHOTS OF MY CODE FROM VISUAL STUDIO CODE USING THE SNIPPING TOOL, WHICH ANSWERS THE QUESTIONS 1, 2, 3, 4, AND 5 RESPECTIVELY.