**LAB 2:** Cassandra Cabrera and Mike Menendez **Due:** Jan. 29, 2020

**Summary:** We both have previous experience converting hexadecimal so this assignment was a good refresher. Also allowed us to learn how these hexadecimals easily transpire into RGB tuples.

**Our Code:**

```
#Adding async support
import asyncio


async def task_master():
    Task()


class Task:
    # Member variables of the Task object
    task, red, green, blue, hex = "", 0, 0, 0, 0

    # Init the passed Task object's RGB members
    @staticmethod
    def rgb_setup(self):
        fmt = False
        while not fmt:
            try:
                print("Please enter the RGB tuple with each color separated with the return
carriage")
                self.red, self.green, self.blue = int(input()), int(input()), int(input())
                fmt = True
            except:
                print("Invalid format, please try again")

    # Init the passed Task object's hex member
    @staticmethod
    def hex_setup(self):
        fmt = False
        while not fmt:
            try:
```

```
            print("Please enter the hexadecimal color string in standard format (ex:
#FFFFFF)")
            self.hex = input().strip()[1:]
            fmt = True
        except:
            print("Invalid format, please try again")

    # Fulfills task1 of the assignment
    # Checks if a RGB tuple is a primary color
    @staticmethod
    def task1(self):
        if self.green < self.red > self.blue:
            print("The color is reddish")

        elif self.blue < self.green > self.red:
            print("The color is greenish")

        else:
            print("The color is blueish")

    # Fulfills task 2 of the assignment
    # Checks if a RGB tuple is a secondary color
    @staticmethod
    def task2(self):
        if self.red == self.blue:
            print("The color is a shade of magenta")

        elif self.red == self.green:
            print("The color is a shade of yellow")

        elif self.blue == self.green:
            print("The color is a shade of cyan")

    # Fulfills task 3 of the assignment
    # Converts hex string to RGB tuple
    @staticmethod
    def task3(self):
        dic = {"A":10, "B":11, "C":12, "D":13, "E":14, "F":15}
        if(self.hex[0].isnumeric() != True):
```

```
            hex0 = dic.get(self.hex[0])
        else:
            hex0 = self.hex[0]
        if(self.hex[1].isnumeric() != True):
            hex1 = dic.get(self.hex[1])
        else:
            hex1 = self.hex[1]
        if(self.hex[2].isnumeric() != True):
            hex2 = dic.get(self.hex[2])
        else:
            hex2 = self.hex[2]
        if(self.hex[3].isnumeric() != True):
            hex3 = dic.get(self.hex[3])
        else:
            hex3 = self.hex[3]
        if(self.hex[4].isnumeric() != True):
            hex4 = dic.get(self.hex[4])
        else:
            hex4 = self.hex[4]
        if(self.hex[5].isnumeric() != True):
            hex5 = dic.get(self.hex[5])
        else:
            hex5 = self.hex[5]
        tup1 = (int(hex0) * 16)+(int(hex1))
        tup2 = (int(hex2) * 16)+(int(hex3))
        tup3 = (int(hex4) * 16)+(int(hex5))
        rgb = (tup1,tup2,tup3)
        print(rgb)

    # Fulfills task 4 of the assignment
    # Converts RGB tuple to hex
    @staticmethod
    def task4(self):
        dic = {10:"A", 11:"B", 12:"C", 13:"D", 14:"E", 15:"F"}
        hex0 = int(self.red/16)
        hex1 = self.red - (hex0*16)
        hex2 = int(self.green/16)
        hex3 = self.green - (hex2*16)
        hex4 = int(self.blue/16)
```

```
        hex5 = self.blue - (hex4*16)

        if(hex0 > 9):
                hex0 = dic.get(hex0)
        if(hex1 > 9):
                hex1 = dic.get(hex1)
        if(hex2 > 9):
                hex2 = dic.get(hex2)
        if(hex3 > 9):
                hex3 = dic.get(hex3)
        if(hex4 > 9):
                hex4 = dic.get(hex4)
        if(hex5 > 9):
                hex5 = dic.get(hex5)

        print("#",hex0,hex1,hex2,hex3,hex4,hex5,sep="")

    # Given any RGB tuple, can determine the hue of the tuple whether it be a primary or
secondary color
    @staticmethod
    def task5(self):
        if self.red == self.green or self.red == self.blue or self.blue == self.green:
            self.task2(self)
        else:
            self.task1(self)

    # Init of the Task object and calls helper functions to handle the specified task
    def __init__(self):
        super().__init__()
        valid = False
        while not valid:
            try:
                print("Please enter task number:")
                self.task = int(input())
                valid = True
            except:
                print("Invalid format, please try again")

        if self.task == 1:
```

```
            self.rgb_setup(self)
            self.task1(self)

        elif self.task == 2:
            self.rgb_setup(self)
            self.task2(self)

        elif self.task == 3:
            self.hex_setup(self)
            self.task3(self)

        elif self.task == 4:
           self.rgb_setup(self)
           self.task4(self)

        elif self.task == 5:
            self.task5(self)
        else:
            print("Error, invalid input")
            exit(69)


# Creates the main Task object which handles processing of a given RGB tuple or Hex
color string
# noinspection PyTypeChecker
async def main():
    await asyncio.create_task(task_master())


if __name__ == '__main__':
    asyncio.run(main())
    exit(0)
```