# EXAM 2 REVIEW

# Question 1

- Given the following class constructor, write a mutator method for its radius.

```
class Balloon:
    def __init__(self, color, radius):
        # color must be a valid RGB color
        self.color = color
        # radius must be between 0 and 12 inclusive
        self.radius = radius

    def setRadius(self, value):
        # YOUR CODE HERE
```

# Question 1

- Given the following class constructor, write a mutator method for its radius.

```
class Balloon:
    def __init__(self, color, radius):
        # color must be a valid RGB color
        self.color = color
        # radius must be between 0 and 12 inclusive
        self.radius = radius

    def setRadius(self, value):
        if value < 0:
            self.radius = 0
        elif value > 12:
            self.radius = 12
        else:
            self.radius = value
```

# Question 2

- Given the following class constructor, write an accessor method for its radius.

```
class Balloon:
    def __init__(self, color, radius):
        # color must be a valid RGB color
        self.color = color
        # radius must be between 0 and 12 inclusive
        self.radius = radius

    def getRadius(self):
        # YOUR CODE HERE
```

# Question 2

- Given the following class constructor, write an accessor method for its radius.

```
class Balloon:
    def __init__(self, color, radius):
        # color must be a valid RGB color
        self.color = color
        # radius must be between 0 and 12 inclusive
        self.radius = radius

    def getRadius(self):
        return self.radius
```

# Question 3

- Consider this function:

```
def mystery(x):
    if x % 2 == 1:
        if x**3 != 27:
            x = x + 4
        else:
            x = x / 1.5
    else:
        if x <= 10:
            x = x * 2
        else:
            x = x - 2
    return x
```

- What happens with the following two statements?

```
print(mystery(8))
print(mystery(5))
```

# Question 3

- Consider this function:

```
def mystery(x):
    if x % 2 == 1:
        if x**3 != 27:
            x = x + 4
        else:
            x = x / 1.5
    else:
        if x <= 10:
            x = x * 2
        else:
            x = x - 2
    return x
```

- What happens with the following two statements?

```
print(mystery(8))
print(mystery(5))
```

- 16
- 9

# Question 4

- Consider the following two implementations of this function:

```
def audioReverse(audio):
    return audio.reverse()

def audioReverse(audio):
    return list(reversed(audio))
```

- Which one preserves immutability? Why?

# Question 4

- Consider the following two implementations of this function:

```
def audioReverse(audio):
    return audio.reverse()

def audioReverse(audio):
    return list(reversed(audio))
```

- Which one preserves immutability? Why?
  - The second one. The first implementation changes the values in the audio list, which may cause problems in later code that depends on it. The second one creates a new list of reversed values and does not modify audio.

# Question 5

- Consider the following problem description – what are the candidate classes?

  You are painting all the walls in your new house. You need to know how many gallons of paint and how much blue painter's tape to buy at the store. You decide to write a program to calculate this for you. Your program will read in data about each wall in your house as well as about any windows and doors on each wall. From this, it will calculate the area to be painted (the wall area minus the windows and doors) and the total length of tape you need to mask the outside of the windows and doors.

# Question 5

- Consider the following problem description – what are the candidate classes?

  You are painting all the walls in your new house. You need to know how many gallons of paint and how much blue painter's tape to buy at the store. You decide to write a program to calculate this for you. Your program will read in data about each wall in your house as well as about any windows and doors on each wall. From this, it will calculate the area to be painted (the wall area minus the windows and doors) and the total length of tape you need to mask the outside of the windows and doors.

- Since paint and tape are simply numbers being calculated, they likely wouldn't be modeled as classes in your program. And house is not really something we need to keep track of, nor is store.

# Question 6

- Assuming we decided we will model walls, windows and doors as objects from the previous code, what attributes do each need to know, and what behaviors do they need to perform?

| Wall | Window | Door |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

- Add on question: How would you structure the inheritance hierarchy?

# Question 6

- Assuming we decided we will model walls, windows and doors as objects from the previous code, what attributes do each need to know, and what behaviors do they need to perform?

| Wall | Window | Door |
|---|---|---|
| Height | Height | Height |
| Width | Width | Width |
| | | |
| Calculate area | Calculate area | Calculate area |
| | Calc. perimeter | Calc. perimeter |
| | | |
| | | |

- Add on question: How would you structure the inheritance hierarchy?

# Question 7

- What are some of the style issues with this code?

```
def move(self, a, b, c):
    if c == 'a' and self.x - 1 >= 0:
        self.x -= 1
        return True
    if c == 'd' and self.x + 1 < b:
        self.x += 1
        return True
    if c == 's' and self.y - 1 >= 0:
        self.y -= 1
        return True
    if c == 'w' and self.y + 1 < a:
        self.y += 1
        return True
    return False
```

# Question 7

- What are some of the style issues with this code?

```
def move(self, a, b, c):
    if c == 'a' and self.x - 1 >= 0:
        self.x -= 1
        return True
    if c == 'd' and self.x + 1 < b:
        self.x += 1
        return True
    if c == 's' and self.y - 1 >= 0:
        self.y -= 1
        return True
    if c == 'w' and self.y + 1 < a:
        self.y += 1
        return True
    return False
```

Poorly named variables. No explanation of what the characters are, no comments on what the code does.