

## EXAM 2 (PRACTICE)

*February 14**Spring 2020*

## Instructions:

- Partial answers will receive partial credit. Make every effort to at least put something for each question. If you don't know the answer, you can get plenty of partial credit by giving a partial answer and explaining where it is wrong or where you got stuck.
- If you believe a problem is incorrectly or incompletely specified, make a reasonable assumption and solve the problem. The assumption should not result in a trivial solution.
- In all cases, clearly state any assumptions you make in your answers.

<b>Name</b>	
-------------	--

Part	Description	Points Possible	Grade
1	Multiple Choice	15	
2	Debugging	10	
3	Coding	20	
Total		50	

## Multiple Choice and True/False Questions

1. [3 points] What is the size, in bytes, of a single RGB pixel?

(a) 1 byte  
(b) 3 bytes  
(c) 256 bytes  
(d) 768 bytes  
(e) None of the above.

Your Answer \_\_\_\_\_

2. [3 points] Which of the following decreases the amount of red in a pixel named `p` by 20%?

(a) `newRed = getRed(p) * 0.2`  
(b) `newRed = getRed(p) * 0.8`  
(c) `setRed(p, getRed(p) * 0.2)`  
(d) `setRed(p, 0.8 * getRed(p))`

Your Answer \_\_\_\_\_

3. [3 points] Let's say I want to save space in my encoding for pixels. So I decide to use just 4 bits to represent each color. How many shades of red can I represent with these 4 bits?

(a) 4  
(b) 8  
(c) 16  
(d) 256

Your Answer \_\_\_\_\_

4. [3 points] The negative of a color is what you get if you subtract the color values from 255.

(a) True  
(b) False

Your Answer \_\_\_\_\_

5. [3 points] Using our standard RGB pixel encoding, how many bytes would be used to store a 500 x 500 image?

(a) 500 bytes  
(b)  $500 + 500 = 1000$  bytes  
(c)  $500 \times 3 = 1500$  bytes  
(d)  $(500 + 500) \times 3 = 3000$  bytes  
(e)  $500 \times 500 = 250,000$  bytes  
(f)  $500 \times 500 \times 3 = 750,000$  bytes  
(g) None of the above.

Your Answer \_\_\_\_\_

## Debugging

6. Bob wanted to design a function that reduces the blue by 40%, but only in the top half of an image. He has run into some problems and brought his code to Alice for help. You can see it below:

```
def reduceBlueTopHalf(picture):  
  
    for px in range(0, len(pixels)):  
  
        oldBlue = getBlue(px)  
  
        setBlue(px, oldBlue * 0.4)
```

- (a) [3 points] Circle the errors, including typos, syntax errors, and logical errors (things that run but produce unintended results) in Bob's code. The extra space between the lines is intentional and meant to provide you with space for your circles; do not circle it.
- (b) [7 points] Write a corrected version of Bob's code. You can change whatever you need or want to change so long as it produces Bob's desired outcome.

## Coding

7. The next three problems go together:

(a) [5 points] Write a function that converts an image to grayscale.

(b) [5 points] Write a function that increases the blue in every pixel by 30 points.

(c) [5 points] Write a function that combines the effect of the previous two functions, i.e., it makes an image grayscale and increases blue by 30 points. For full credit you must employ hierarchical decomposition and reuse the functions from the previous two questions.

8. [10 points] Write a function that will decrease red to 0 in the bottom half of the pixels of an image.

9. [10 points] Write a function that negates the middle third of an image.

10. [10 points] Write a function that mirrors the top half of an image. This means taking the first pixel and swapping it with the last in the top half, the second with the second to last in the top half, and so forth.