

## Chapter 5

### Code Changes and Expected Failures:

We injected three errors into the color converter class. The first was in the method `hex2Rgb(Color color)`. Within the print statement we changed the order from red, green, blue to blue, red, green. This method is used in test cases 6 to 10 and we expected tests 6, 9, and 10 to fail, but tests 7 and 8 to pass. The reasoning for this is that 7 has a correct output of `rgb(0, 0, 0)` and 8 has a correct output of `rgb(255, 255, 255)`, so the order does not matter.

The second fault was injected into the method `rgb2Hex(Color color)`. The fault is similar to the one in `hex2Rgb`, we changed the color order from red, green, blue to green, blue, red. This method is correlated to test cases 1 to 5 and we expected similar results such that 1, 4, and 5 fail and 2 and 3 stay passing as their results are `#000000` and `#FFFFFF` so order does not matter.

The third fault was to the method `offsetRgbColor(Color bgcolor, int offsetRed, int offsetGreen, int offsetBlue)`. Instead of adding each offset color int to the corresponding rgb values of `bgcolor`, the values are now subtracted. This method is for test cases 16 through 20 and we expected all to fail due to that change.

The final two faults were injected into the contrast checker class. We changed the static variable `CONTRAST_FACTOR` from 0.05 to 0.5. This change affects the method `computeContrast(Double lighter, Double darker)`. We made a change within the `computeContrast` method as well. Instead of having the value of the lighter be divided by the darker, they are now multiplied with each other. This method is associated with test cases 11 through 15 and we expected them all to fail after these changes.

We did not want to change the method used in test cases 21 through 25, since several of those test cases failed due to an error already existing within the code.

The results are as expected and shown below.

## Team CargoPants

Isabel Lally

Collin Bauer

Dylan Evans

Mon Nov 18 21:29:11 EST 2019

Case	Requirement	Method	Inputs	Output	Oracle	Result
01	The method will take a valid color defined by rgb and change it to hex.	ColorConverter .rgb2Hex()	220 136 15	#880FDC	#DC880F	fail
02	The method will take a valid color defined by rgb and change it to hex.	ColorConverter .rgb2Hex()	0 0 0	#000000	#000000	pass
03	The method will take a valid color defined by rgb and change it to hex.	ColorConverter .rgb2Hex()	255 255 255	#FFFFFF	#FFFFFF	pass
04	The method will take a valid color defined by rgb and change it to hex.	ColorConverter .rgb2Hex()	30 15 2	#0F021E	#1E0F02	fail
05	The method will take a valid color defined by rgb and change it to hex.	ColorConverter .rgb2Hex()	184 93 25	#5D19B8	#B85D19	fail
06	The method will take a valid color defined by a hex value and convert to rgb.	ColorConverter .hex2Rgb()	0xDC880F	rgb(15, 220, 136)	rgb(220, 136, 15)	fail
07	The method will take a valid color defined by a hex value and convert to rgb.	ColorConverter .hex2Rgb()	0x000000	rgb(0, 0, 0)	rgb(0, 0, 0)	pass
08	The method will take a valid color defined by a hex value and convert to rgb.	ColorConverter .hex2Rgb()	0xFFFFFFFF	rgb(255, 255, 255)	rgb(255, 255, 255)	pass
09	The method will take a valid color defined by a hex value and convert to rgb.	ColorConverter .hex2Rgb()	18 52 86	rgb(86, 18, 52)	rgb(18, 52, 86)	fail
10	The method will take a valid color defined by a hex value and convert to rgb.	ColorConverter .hex2Rgb()	0xAABBCC	rgb(204, 170, 187)	rgb(170, 187, 204)	fail
11	This method takes two doubles and computes the contrast between them and outputs in the form of a double.	ContrastChecker .computeContrast()	50.0 20.0	1035.25	2.5	fail
12	This method takes two doubles and computes the contrast between them and outputs in the form of a double.	ContrastChecker .computeContrast()	-50.0 0.0	-24.75	-999.0	fail

	double.					
13	This method takes two doubles and computes the contrast between them and outputs in the form of a double.	ContrastChecker .computeContrast()	100 250	25175.25	0.4	fail
14	This method takes two doubles and computes the contrast between them and outputs in the form of a double.	ContrastChecker .computeContrast()	100.0 100.0	10100.25	1.0	fail
15	This method takes two doubles and computes the contrast between them and outputs in the form of a double.	ContrastChecker .computeContrast()	1000000.9 75.5	7.60001064E7	13236.28	fail
16	This method takes six integers, the first three represent the rgb values of a color	ColorConverter .offsetRgbColor()	250 0 0 3 10 15	Color parameter outside of expected range: Green Blue	java.awt.Color[r=255,g=10,b=15]	fail
17	This method takes six integers, the first three represent the rgb values of a color	ColorConverter .offsetRgbColor()	100 150 255 -50 -100 -200	Color parameter outside of expected range: Blue	java.awt.Color[r=50,g=50,b=55]	fail
18	This method takes six integers, the first three represent the rgb values of a color	ColorConverter .offsetRgbColor()	2 5 16 43 101 99	Color parameter outside of expected range: Red Green Blue	java.awt.Color[r=47,g=106,b=113]	fail
19	This method takes six integers, the first three represent the rgb values of a color	ColorConverter .offsetRgbColor()	101 101 202 25 50 -101	Color parameter outside of expected range: Blue	java.awt.Color[r=126,g=151,b=101]	fail
20	This method takes six integers, the first three represent the rgb values of a color	ColorConverter .offsetRgbColor()	50 50 50 50 50 50	java.awt.Color[r=0,g=0,b=0]	java.awt.Color[r=100,g=100,b=100]	fail
21	This method calculates the distance between two colors using 3-dimensions Euclidean distance.	DistanceCalculator .calculate()	255 255 255 0 0	367.77	441.67	fail
22	This method calculates the distance between two colors using 3-dimensions Euclidean distance.	DistanceCalculator .calculate()	200 100 10 200 100 10	0.0	0.0	pass
23	This method calculates the distance between two colors using 3-dimensions Euclidean distance.	DistanceCalculator .calculate()	0x000000 0xFFFFFFFF	367.77	441.67	fail
24	This method calculates the distance between two colors using 3-dimensions Euclidean distance.	DistanceCalculator .calculate()	0xAABBCC 0xAABBCC	0.0	0.0	pass
25	This method calculates the distance between two colors using 3-dimensions Euclidean distance.	DistanceCalculator .calculate()	20 80 140 10 80 170	29.62	31.62	fail

**Experiences and Lessons Learned:**

The faults were simple enough to create and the results were as expected. We did not have any roadblocks working with this deliverable. Several of the faults we used were changes in the formulas that were used, which is interesting because that is the same type of fault that caused the code to fail in our initial tests. We learned that it is quite simple to mess up code and it is understandable that there are mistakes in the code, but that is why testing is so important.