Chapter 5

Experiences

For this task we injected 5 faults into our methods that we are testing. We put three faults in ColorCoverter.java and one fault in to ColorNameLookup.java and ContrastChecker.java. The faults below do not make all test fail. They cause six test cases to fail, Test Cases 01, 11, 14, 15, 21, and 25. Injecting these faults into the code showed us that just little discrepancies and errors in the logic of the code can cause issues with the functionality of the program.

Faults Added

ColorConverter.java Faults

```
1. //FAULT ADDED - Changed max angle to 180
   private static final int MAX_ANGLE = 180;
2. //FAULT ADDED- Changed the max letter to E
   private static final String HEXADECIMAL_DICTIONNARY = “[0-9A-Ea-
f]+";   //  FFF,  FFFFFF
3. //FAULT ADDED- Removed toLower Case command from string str
   public static Color colorFromRgbStr(String colorStr) {
   Color  color = null;
   String str   = colorStr.replaceAll("\\s", ""); // replace ' ', \t,
\n, ...
   if (str.matches(RGB_DICTIONNARY) |
str.matches(SHORT_RGB_DICTIONNARY)){  // ex: rgb(255,255,255) or
255,255,255
            str = str.replaceAll("rgb\\(", "");
            str = str.replaceAll("\\)", "");
            String[] strList = str.split(",");
            int r = Integer.parseInt(strList[0]);
            int g = Integer.parseInt(strList[1]);
            int b = Integer.parseInt(strList[2]);
            if(     r <= RGB_MAX && g <= RGB_MAX && b <= RGB_MAX
                &&  r >= RGB_MIN && g >= RGB_MIN && b >= RGB_MIN){
                color = new Color(r, g, b);
            }
        }
        return color;
    }
```

ColorNameLookUp.java Faults

```
1. //FAULT ADDED- Yellow changed to yellow
   private ArrayList<ColorName> initColorList() { …
   colors.add(new ColorName("yellow", 0xFF, 0xFF, 0x00));
   …}
```

ContrastChecker.java Faults

```
1. //FAULT ADDED- the green value is being cubed rather than squared
public static double distanceColor(final Color fgColor, final Color
bgColor) {
        int redFg = fgColor.getRed();
        int redBg = bgColor.getRed();
        int greenBg = bgColor.getGreen();
        int greenFg = fgColor.getGreen();
        int blueFg = fgColor.getBlue();
        int blueBg = bgColor.getBlue();
        return (Math.sqrt(Math.pow(redFg - redBg, 2) +
Math.pow(greenFg - greenBg, 3) + Math.pow(blueFg - blueBg, 2)));
    }
```