

Assignment #1 – Color in Java using IntelliJ

Date assigned: Monday, January 8, 2018

Date due: LearnYourColors Design Tues, Jan 9, 2018 by 1pm (10 pts)
LearnYourColors Text Implementation, Thurs, Jan 11, 2018 by 1pm (20 pts)
LearnYorColors Android Implementation, Fri, Jan 12, 2018 by 1pm (30 pts)

Over the next week, we are going to develop two LearnYourColors applications. One application is going to be developed using IntelliJ IDEA and is text-based. The other application is going to be developed using Android Studio 3.0.1 and is graphical. Your job is to carefully design a codebase that can be maintained and used in both applications.

ANSI colors

Any text-based application will use ANSI colors as follows:

```
“\u001B[30m” Black  
“\u001B[31m” Red  
“\u001B[32m” Green  
“\u001B[33m” Yellow  
“\u001B[34m” Blue  
“\u001B[35m” Purple  
“\u001B[36m” Cyan  
“\u001B[37m” White
```

The statements:

```
System.out.println (“\u001B[34m” + “CS”);  
System.out.println (“\u001B[35m” + “IS”);  
System.out.println (“\u001B[32m” + “FUN”);
```

produces the output:

```
CS  
IS  
FUN
```

An ANSI color is to associate the color name (e.g. Black) and the ANSI string color (e.g. “\u001B”).

ARGB colors

Any of our Android applications will use ARGB colors as follows:

```
“0xff000000” Black  
“0xffff0000” Red  
“0xff00ff00” Green  
“0xffffffff00” Yellow  
“0xff0000ff” Blue  
“0xff800080” Purple  
“0xff00ffff” Cyan  
“0xffffffff” White
```

An ARGB color is to associate the color name (e.g. Black) and the ARGB string color (e.g. "0xff000000"). The key thing here is that you need to take the string color "0xff000000" and turn it into an int 0xff000000. This might not be as easy as you think. You are to provide proper behavior for a ColorARGB object.

An example of how the ARGB value will be used in Android is:

```
int aColor = 0xff0000ff; // alpha(ff) fully opaque, red(00), green(00), blue (ff)
```

...

```
anEditWidget.setBackgroundColor (aColor); // this method takes an int as an ARGB value
```

You are to create a Java project **PUNetIDColor** that creates a text-based application that helps a young gradeschooler learn some basic colors. The application runs the user through all of the ANSI colors keeping score as the application is run. At the end, if the user gets all colors correct, print "Great Job"; otherwise, print "Keep Trying"

Here is a sample output using a seed of 0 to seed a Random object such as `mRandomGenerator = new Random (seed);` The method `nextInt` gets a pseudo-random number between `[0, value)` where `value` is the argument to `nextInt`.

Figure 1:

```
Learn Your Colors
```

```
Color? Purple
Score: 1 / 1
```

```
Color? Blue
Score: 1 / 2
```

```
Color? Red
Score: 2 / 3
```

```
Color? Purple
Score: 2 / 4
```

```
Color? Green
Score: 3 / 5
```

```
Color? Black
Score: 4 / 6
```

```
Color? Brown
Score: 4 / 7
```

```
Color? Gray
Score: 4 / 8
```

```
Keep Trying
```

LearnYourColors Design

You are to design the entire LearnYourColors text-based application using UMLet. Your design is to be named PUNetIDLearnYourColors.uxf. The text-based application does not use any of the ARGB information I described; however, your design is to include ARGB behavior for the Android

application. The main difference between ANSI and ARGB colors is that ANSI colors will only use <name, value> pairs as Java Strings. ARGB colors will also have <name, value> pairs as Java Strings but will have additional behavior to be able to return the String value as an int. All I/O is to happen in the method main. None of the other classes are to have any I/O.

On the due date, print and turn in a one-page copy of your design identifiable to you. Also, drop the specified file into the CS260-01Drop folder on grace.

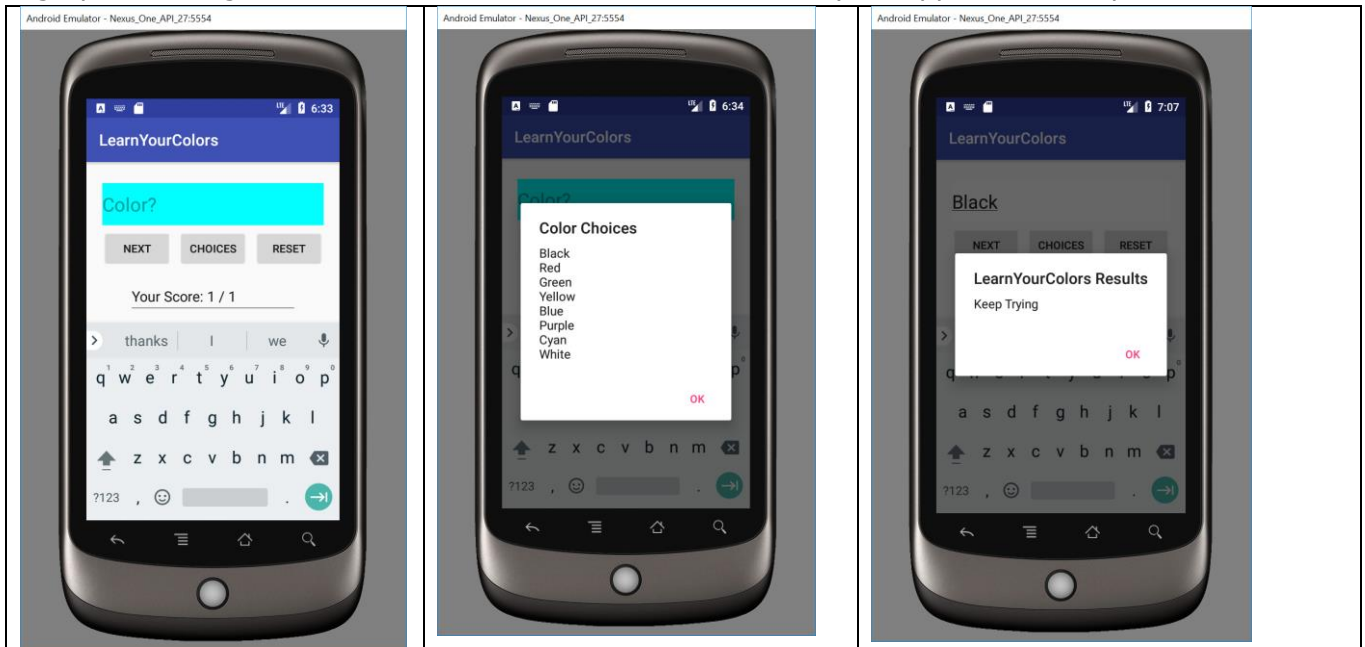
LearnYourColors Text Implementation

This is the Java text-based implementation of your design to run the application as shown in Figure 1 above. As you are implementing your Color class, I want you to write a ColorTest JUnit3 test class. You do not have to write any other test classes although you are advised to do so. In a perfect world with unlimited time, each class would have a test class, but we are on limited time, so I just want to make sure you know how to do this.

On the due date, print and turn in each class starting with the class containing main. Print a superclass before any class that inherits from said superclass. Also, drop the entire project into the CS260-01Drop folder on grace.

LearnYourColors Android Implementation

The Android implementation will work similarly to the text-based version; however, the application is graphical using the Android API. Here are the screen shots your application will produce:



On the due date, print and turn in each new Android class starting with the class containing the MainActivity. Do not turn in any classes from the text-based application; however, turn in any new classes used for the Android application. Also, drop your entire Android project used to create the Android application just described into the CS260-01Drop folder. We will go over more of this in class as the week progresses.

Goals for Assignment #1:

1. Write a Java application using multiple classes
 2. Understand the concept of packages to better organize classes
 3. Use good OOP techniques in implementing your solution
 4. Use the Java API which has a rich library of routines (e.g. Vector, Stack)
 5. Use JUnit framework for testing classes
 6. Use the IntelliJ IDEA for project development and testing
 7. Write an Android application reusing some classes from the text-based version
 8. Use Android Studio 3.0.1 for project development and testing
-

Specifics:

1. If you come to me with a question regarding your solution, I will have you load your project onto a machine in the CS lab. I will not look at your code on your computer or on paper as it just takes me too long to get at the problem. Further, I want you to bring in your lecture notes in case I want you to look up something. Remember, I'm not just a tell you the answer guy. Make sure you understand how to use the developer tools. You need to be proficient using the debugger, so start using the debugger ASAP.
2. If you want help with a compiler error, you must be able to tell me exactly what statement you put in your code that caused the error and be able to isolate the error. If you have typed in a bunch of code and have not tested your code as you've gone along, I'm not going to help you sort out the mess. You've been warned!!
3. We will be in class approximately 3 hours per day. You are expected to spend a minimum of 6 hours outside of class time per day ... EVERY DAY ... including the first day of class. This is just like a job!!!! ☺ ☺