CSCI 2540 Assignment 1

60 points

Due date: Tuesday, Sept. 6, 11:59pm

****

Instructions for setting up your program structure in Eclipse:

If you have not already created a Java Project called "CSCI2540", create one first.

Right click "CSCI2540" and create a new package for this assignment. Name the package as assg1_YourPirateId.

Right click the package and create a new class for each problem in this assignment.

****

1. (20 pts) Create a new Java class inside your package. The name of the class should be **ColorFinder.**
   (Note that the name of your file should be **ColorFinder.java)**

   In color theory, the primary colors of an additive color system are red, green, and blue. An example of a computer device which uses the additive color system would be your computer monitor, because it adds red, green, and blue light together to make all of the colors you see on your screen.

   The primary colors of a subtractive color system are magenta, yellow, and cyan. An example of a computer device which uses the subtractive color system would be your printer, because it mixes magenta, yellow, and cyan colored inks together to make all of the colors you see on paper.

   Write a program that gets a string from the keyboard and tests whether the string contains one of the primary colors.

   Your program should be able to find the primary colors, no matter what letters in the color are capitalized. For example, your program should be able to finds words like **grEeN**.

   Begin by asking the user to type in a sentence. Once you have the sentence, you should proceed to test it to see if it contains one of the primary colors.
   If one of the additive colors is found in the sentence, print the message "**Additive primary color found.**" to the screen.
   If one of the subtractive colors is found in the sentence, print the message "**Subtractive primary color found.**" to the screen.
   Preference should be given to the additive colors, so that if a sentence contains both additive and subtractive colors, you should ONLY print the message "**Additive primary color**

found." to the screen.
If none of the colors are found in the sentence, print the message "**No primary colors found.**" to the screen.

The following is an example of what you **MIGHT** see on the screen when your program runs. The exact output depends on the values that the user types in while the program runs. The user's inputted values are shown below in italics:

Enter a sentence:
Roses are red, violets are blue.

Additive primary color found.

Here is another example run:

Enter a sentence:
Aqua is a color that is similar to Cyan.

Subtractive primary color found.

Here is another example run:

Enter a sentence:
Have you been to Yellowstone Park?

Subtractive primary color found.

Here is another example run:

Enter a sentence:
Yellow is next to green in the rainbow.

Additive primary color found.

Here is another example run:

Enter a sentence:
Black is technically not a color.

No primary colors found.

2. (20 pts) Create a new JAVA class in your package. The name of your new JAVA class should be **NumberGame**.

   Write a Java program that plays a number guessing game with the user.

   Your program should begin by choosing a random number between 0 and 99 (inclusive). This can be done with the following statements:

   ```
   int secret;
   Random generator = new Random();
   secret = generator.nextInt(100);
   ```

   After executing these statements, the variable *secret* will be holding a randomly generated number from 0 to 99. If you need to get another randomly generated number, you just simply repeat the last line again.

   Next, your program should ask the user to try to guess what number was chosen.

   If the user does not guess the number correctly, you should tell the user that the secret number is higher or lower than the one they guessed. At this point you should allow them to guess again.

   Repeat this process until the user guesses the number correctly.

   Whenever the user does correctly guess the number, you should stop the program and print to the screen the number of tries that it took the user to guess the number.

   Hint for Testing Your Program:

   While testing your program, you can simply set *secret* to a known value, so that you can properly test your program by knowing the correct answer.

   For example:

   ```
   int secret;
   Random generator = new Random();
   secret = generator.nextInt(100);
   secret = 54;
   ```

   Once you know the program works correctly, you need to take the “secret = 54;” line out so the answer will be random each time the program executes.

3. (20 pts) Create a new JAVA class for this assignment. The name of your new JAVA class should be **CountFamilies**.

   Write a program that counts the number of families whose income is below a certain level.

   Begin your program by asking the user to enter the number of families. Store this value in a variable called numOfFamilies.

   Next, create an array of size numOfFamilies. Then you will read numOfFamilies values representing family incomes from the keyboard and place them into the array.

   Next, find the maximum income of all the values entered, and print this value to the screen.

   Finally, count the families that make less than 10% of the maximum income. Display the income of each of these families, then display the count.

   For example:

   Please enter the number of families: *5*

   Enter an income: *8500*
   Enter an income: *109000*
   Enter an income: *49000*
   Enter an income: *9000*
   Enter an income: *67000*

   The maximum income is: 109000

   The incomes of families making less than 10% of the maximum are:
   8500
   9000
   for a total of 2 families

   ****
   Technical Notes
   - At the top of EVERY Java file you create, you must include a comment which states your name. Include other commends in your programs as needed.
   - Programs that do not compile will receive an automatic grade of "F".

   Submission instructions:

   You need to submit your programs electronically on Canvas.

   Please **use a named package for each of your assignment.** For example, for assignment 1, create a new package and **name your package as assg1_yourPirateId** (use lower case for your pirate id), such as assg1_smithj20. You also need to include a statement such as "package assg1_smithj20;" at the beginning of each of your .java file (it will be automatically generated in Eclipse if you create the class inside the given package). **Please follow this naming convention exactly for all future assignments. You will be deducted points for not doing so.**

When you submit your files to Canvas, please submit **a zip file with your package folder inside the zip file.** The package folder should **include only .java files (make sure you include .java files, not .class files).** The name of the folder should match with your package name. (You can use 7-zip software to zip files/folder). Once your zip file is unzipped, it will generate a folder that matches your package name.

A file called "*Instruction on submitting an assignment*" is also posted on Canvas (under "Getting Started" module).

****