**CS 399: Mobile Application Development**

**Summer 2020**

**Homework-08  
Due Friday June 26**

**30 Points Total**

#### **Instructions:**

1. This is an individual assignment. It is okay to exchange ideas with other people or review external material on the assignment topics but the work you submit should be your work.
2. Include a header in your assignment sheet including your name, the section number, and the homework number.
3. Cite any external sources you use to help you with your assignment.
4. Submit your assignment on the assignment page on BBLearn. Submissions via email will not be accepted.

***Android Application Requirements:***

Implement a Flutter mobile application modeled based on the *Dice App / Dicee Project* application implemented in the *Flutter: Part4* LinkedIn training. This project is an extension of the project you completed for Homework-07. Requirements for the application are listed below:

1. The application will have a single screen displaying with five die images and one textBox.
2. Use *Scaffold* widget for the basic material design structure of your application screen display.
3. Include an *AppBar* widget for the screen display and set its title to a name representative of the application purpose/operations.
4. The application at the start shows five die with the die images for side one.
5. The application at the start shows ROLLEM in the textbox on the display screen.
6. Clicking on any die image will roll all five die.
7. After dice roll the application’s screen displays the five dice after rolling.
8. After dice roll the application’s screen displays one of the following messages in a textbox based on the dice roll outcome:
   1. No Particular combination
   2. 2 Pairs
   3. 3 of a kind
   4. 3 of a kind and a pair
   5. 4 of a kind
   6. 5 of a kind
9. Use unique choice of Material colors for the *Scaffold* and *AppBar* widgets *background* colors. Select colors from collection of Material colors available [here](https://material.io/design/color/the-color-system.html#tools-for-picking-colors).
10. Use a custom *app icon* for your application for the iOS and Android phone homepage.
11. Be creative and aim for an elegant and visually appealing screen display.

**Implementation:**

1. Implement your application using Android Studio, Flutter platform and Dart programming language.
2. Start your Flutter application project from your completed project for Homework-07.
3. Create a class (e.g. *DiceRoller*) under your Flutter *lib* directory. Use this class to encapsulate the data representing the state of five die (5 integers for storing dieNumbers for each die), and operations for rolling the dice, reporting the state of the five die and reporting the dice-roll outcome message. Have a constructor for the class that does not have any parameters and sets all 5 dieNumbers to 1. You must import your *DiceRoller* class in the *main.dart* fileto be able to use the class in main.dart. Then create an instance of the *DiceRoller* class in your application and use it to perform the dice operations.

The *References* section below provides reference for instructions for creating a class in a Flutter project.

**Testing:** Use *Pixel 2* or *Nexus 6* virtual device with API level 28 to run your application.

***Deliverables:***

1. Complete Android project files in a zip folder. Name your folder using the naming convention like: *lastname\_firstname\_CS499\_hw8\_Dice.zip*. **Use Android Studio *File* menu and select *Export to Zip File.*** Do not zip the project file directly from the project directory.
2. Include snapshots of the following artifacts as displayed in the Android Studio. Paste the snapshots in PDF file named *lastname\_firstname\_CS399\_hw8\_Snapshots.PDF*.
   1. The Dart files under the *lib* directory. Expand all the code to show the complete Dart code including the *import* instructions.
   2. The Android Studio Project Pane showing any directories and components that you have added/changed for your application. Expand android, images, ios and lib directories to show all application components.
   3. Show at least one android *mipmap* image and one iOS *icon-App* image displayed in the Android Studio Editor pane.
   4. The *pubspec.yaml* file after removing all comments.
3. View of your application when it is run on the *Pixel 2* or *Nexus 6* virtual device.
4. View of Android *Pixel 2* or *Nexus 6* home page including display of your *app ico*n.

***Reference:***

**LinkedIn Learning Video**

Flutter: Part 4 - Building Apps with State

Flutter: Part 6 – Modularizing and Organizing Flutter Code / Creating a question class

* Use the instructions for creating a Dart class in a Flutter project for creating your *dice roller* class.