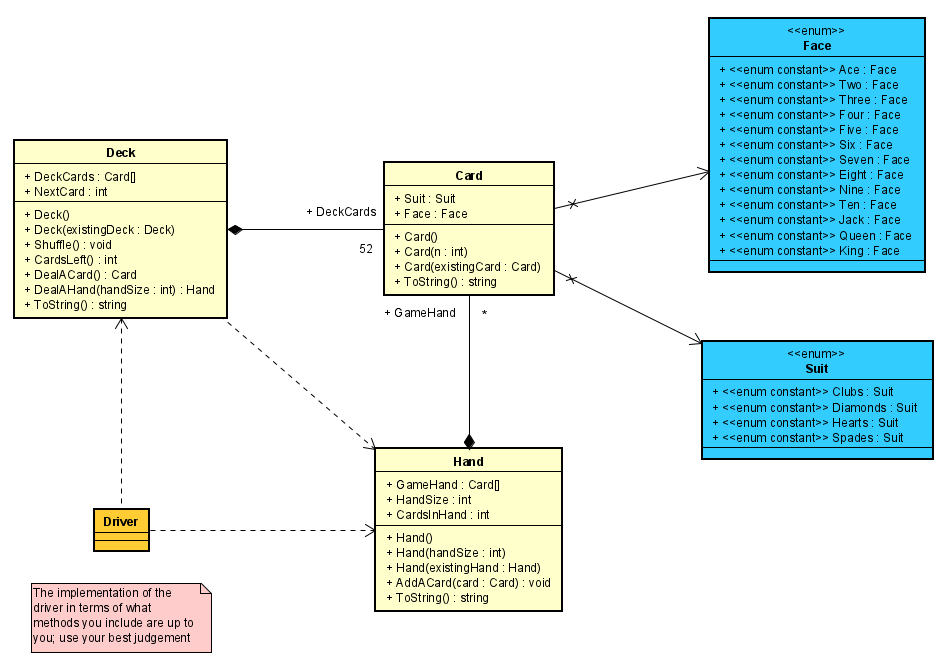


# Project

This project continues work on the project from Lab 2. Be sure to make any needed corrections to Lab 2 before beginning on this assignment.

Extend the functionality from Labs 1 and 2 you have developed previously with the ***Card*** and ***Deck*** classes by adding a ***Hand*** class that implements the following ***UML*** diagram:



Note that in addition to implementing the ***Hand*** class, you will need to modify the ***Deck’s*** ***DealAHand*** method to return a ***Hand*** object instead of a ***string*** object.

Use the same ***Driver*** from Lab 2. *Make sure it retains all the output from Lab 2*. In addition, it should prompt the user for the size of a ***Hand*** in the game being played and for the number of players.

It should then ***Shuffle*** the original ***Deck*** (not the copy deck) and ***Deal*** and display the ***Hands*** for all players unless there are not enough ***Cards*** in the ***Deck*** (for example, we cannot deal 10 ***Hands*** of 7 ***Cards*** from a ***Deck*** of 52 ***Cards***). The ***Driver*** should display an error message if the user attempts to deal in an impossible situation such as this.

# Hints and Other Specifications

1. The ***Hand*** class should allow for a ***Hand*** of any size from 1 to 52 ***Cards***.
2. The ***Hand*** class should have a ***copy*** ***constructor*** that does a ***deep*** ***copy***.
3. The default ***hand*** ***size*** is ***5*** ***Cards,*** but a different size may be specified via the **parameterized constructor**.
4. The ***HandSize*** is the number of ***Cards*** that the ***Hand*** object can hold while the ***CardsInHand*** is the number of ***Cards*** that are currently in the ***Hand***. The ***CardsInHand*** is initially ***0*** until some ***Cards*** are dealt into the ***Hand***, and it increases by one every time a ***Card*** is added (***dealt***) to the ***Hand***. It is always true that ***HandSize* >= *CardsInHand.***
5. The ***AddCard*** method in ***Hand*** can be used to add one ***Card*** at a time to the ***GameHand*** array until it is full. It should increment the ***CardsInHand*** counter each time a ***Card*** is added to the ***Hand*** as long as there is room for the ***Card*** to fit into the ***Hand***.

# Proper Documentation is Required

Make sure **all** code files are [**fully documented**](http://csciwww.etsu.edu/bailes/courses/1260/LectureMaterial/Policies%20Regarding%20Code%20Documentation.docx)– see the [**course documentation policies**](http://csciwww.etsu.edu/bailes/Courses/1260/LectureMaterial/Policies%20Regarding%20Code%20Documentation.docx) posted on D2L for the conventions and requirements including examples of proper documentation. Proper documentation includes following the naming conventions for classes, files, methods, variables, constants, and other identifiers.

# Submission

Your submission should be a single **.zip** file with a name in the format of ***1260-LastFirst-Project2*** containing your entire VS project. Submit the **one** **zipped** **file** to the **Lab 2 Dropbox** on D2L. There is no design document to include for this project.

# Sample Output

The following figure shows sample output from one possible run of the program.

\**The output from the previous lab has been excluded from this sample screenshot. Make sure the output from the previous lab is also displayed.*

