

# ELEC 2543 Object-Oriented Programming and Data Structures

Homework Assignment 1  
Due Date: 9:30am, March 16, 2021 (Tue)

# Overview

To develop a card game that involves:

- shuffling of a deck of cards
- card distribution among a set of players
- card comparisons

# The Game

- There are several players.
- In the beginning of the game, each player takes turn to draw a card from a shuffled deck of 52 cards.
- The players draw as many cards as they can but they all should have the same number of cards.
- For example, if there are 5 players, then each player will get 10 cards.

# The Game

- The game then proceeds in rounds.
- In each round, each player plays a card. The player who plays the largest card wins.
- The order of the cards that a player plays is the order of getting the card.
- For example, if Alice draws S5, D2, HK, SA, C4 in this order, she will play S5 in the first round, D2 in the second round, and so on.

# Implementation

- Classes for representing Rank, Suit, and Card have been developed in Lab 5.
- You have to develop classes **Deck**, **Player**, and **CardGame**.
- Classes **InitializePlayer** and **CardGameDriver** are provided.

# Class Deck

- This class represents a deck of 52 cards. There are at least two methods: `Deck(int n)` and `Card drawCard()`.
- `Deck(int n)` is the constructor. The parameter tells how many rounds the deck should be shuffled.
- In each round of shuffling, the whole deck is first divided into two sub-decks.
- The sub-deck then interlaced into one whole deck. To simplify the discussion, we assume the cards are 1, 2, ..., 10.

# Class Deck

- In the first round, the whole deck is divided into  $[1, 2, 3, 4, 5]$  and  $[6, 7, 8, 9, 10]$ . We then combine the two sub-decks by interlacing them to  $[1, 6, 2, 7, 3, 8, 4, 9, 5, 10]$ .
- In the second round, we again divide the whole decks into two sub-decks  $[1, 6, 2, 7, 3]$  and  $[8, 4, 9, 5, 10]$  and then combine them to  $[1, 8, 6, 4, 2, 9, 7, 5, 3, 10]$ .

# Class Deck

- Card `drawCard()` removes the first card in the deck and returns it.
- Refer to the deck after the second round in the above discussion, `[1, 8, 6, 4, 2, 9, 7, 5, 3, 10]`, `drawCard()` returns 1 and the deck becomes `[8, 6, 4, 2, 9, 7, 5, 3, 10]`.



# Class Player

- This class represents the players. The skeleton has been provided. Develop methods
- `void addCard(Card card)`  
This method simulates the process that the player draws a card from the deck.
- `public Card playCard()`  
This method simulates the process that the player plays a card from his/her hand.

# Class CardGame

- This class defines the game logic.
- Develop method `public void play()` that simulates a single game – from distributing cards among players to the end of the game.
- Messages should be printed out to describe how the game proceeds. Please refer to the sample output for details.

# The Full Application

- `CardGameDriver.java` is the starting point of the application.
- It calls `InitializePlayer.getPlayers()` to get an array of players.
- You can assume there are at least two players but you cannot assume you know the exact number of players.
- A sample output is provided in `output.txt`.