# ****Card Game: Game of War****

Create a card game application and classes with the following capabilities:

* Use Enums for the suit and value of a card
* Include sort() (a hand) and shuffle() (a deck) methods
* Add a dealCard() method Deck class
* *Modify the GroupOfCards to be an abstract class*
* Create a class called GameOfWar that has three methods:
  + initialize()
  + play()
  + displayWinner()
* Create a GameDriver class that asks the user if he/she wants to play a game and then instantiates a GameOfWar object where the actual game is played (see above).
* Use a constant to hold the number of rounds to play. Use this to verify that your application is working correctly.
* Make sure you have a placeholder for the pile of cards that are being compared
* Make sure you have a placeholder for each player’s pile as they win rounds
* Create the methods to play the actual game of War. This should include the following:
  + Splitting the deck (each player should have 26 cards to start)
  + Comparing the top card from each player
  + Determining a winner and putting the cards into the winning player’s pile
  + Handling the scenario where a player (or both) has no more cards in his/their hand
    - At this point the player’s pile should become the player’s hand
    - Make sure that your cards come out in the proper order when you turn the pile over
  + Handling the scenario of War – where the player’s cards are the same value. In this scenario, make sure the following is done:
    - Each player adds their next three cards to the pile and then turns over/compares their fourth card
    - The winner of the fourth card comparison wins all the cards
    - If there is a tie, this scenario continues
  + Playing until the game ends (one player has all the cards) or the number of rounds played => the number of rounds constant
  + Determining who is the winner of the game

# Enhancements for Final Project

* Include a second game “Magic Card Shuffler and Sorter”. This game should do the following:
* Prompt the user for the number of players (you need at least 4).
* Prompt the user for the number of cards in each hand (must be at least 7 cards).
* Generate a group of cards from the deck for each hand. The cards should be shuffled before creating the hands.
* For each hand, you must sort the cards and determine the winner where highest card wins for each card in the hand.
  + Loop through each hand 1 card at a time.
  + Take 1 card off of the sorted hand and compare between the hands.
  + The highest card gets 1 point. Ties each get a point.
* Total the points for the different players. The player with the highest count (number) wins.
* From the GameDriver, prompt the user about which game he/she wants to play. If the user selects “Game of War”, play your Game of War. If the user selects ““Magic Card Shuffler and Sorter”.
* For both games, store the game running in a text file labeled with the game name. These should be text files called “GameOfWarResults.txt” and “ShufflerSorter.txt”

More Annoybullshit at the last fucking minute because the teacher doesn’t know how to manage shit:

To help with the final project here are some things to make sure you have

GroupOfCards should have the following:

A constructor to build an ArrayList of Card objects

A method to remove a card from the ArrayList (i.e. removeCard(int index))

A method to add a card to the ArrayList (e.g. addCard(int index) where

index is the index in the ArrayList for the card)

Getters and setters

Your Deck class should have the following:

A constructor for the deck that uses the GroupOfCards constructor

(super()) and then generates all the cards and adds them to the deck

A shuffle() method that will shuffle the deck of cards. You can use the

Collections.shuffle() method provided by Java

Your Hand class should have the following

A constructor for the deck that uses the GroupOfCards constructor

(super()). It will be an empty ArrayList until you start the game.

GameOfWar class

An initialize() method where you do the following:

Create a deck object and shuffle the deck

Create two hand objects from the Hand class (1 or each player - hand1 and

hand2)

Create two pile objects from the Hand class (1 for each player that holds

the cards as they are played)

Create two winPile object from the Hand class (1 for each player that

holds the cards that the player wins)

Loop through the deck and remove a card at a time and add the card to

hand1 or hand2 until all the cards in the deck have been added to the

hands. One way to do this is to create a helper method called

dealCards().

A play() method where you do the following:

Take the top card from hand1 and add it to pile 1

Take the top card from hand2 and add it to pile 2

Compare the pile1 card to the pile2 card

Depending on the high card, add both the pile1 and the pile2 cards to the

winner's winPile

Contine until one player has no more cards or the number of rounds has

been met