**Deck of Cards:**

Utilize the concepts of association, inheritance, enumeration, and polymorphism to write a complete program that implements the beginning functionality of a set of card games. We will be expanding on this card game in upcoming weeks, so you’ll want to make this as OO as possible. As a starting point, create the following classes and add methods to perform the listed functionality:

A DeckDriver class to run/test your application. We will be expanding this to test a variety of card games

A Card class with the appropriate attributes/variables to track the number and suit of a card.

A GroupOfCards class that is a generic superclass for different groups of cards. Groups of cards should encompass a Deck and a Hand. These should be separate classes with the appropriate variables and methods.

A Game class to actually track the playing of the game. Different games we will be adding in include BlackJack, Hearts, and War.

**Additional requirements:**

* Use an ArrayList to store Card objects.
* Use enumeration to handle the Card suits.
* The Deck constructor should initialize your ArrayList with the 52 cards found in a standard deck. Each card is a Card object. Each Card object contains two instance variables ─ num and suit.
* Code the ability to Shuffle a deck and orderHand to rearrange the cards in a hand from highest to lowest by suit.
* A dealCard method should remove the next card in the deck.
* A toString method for the deck should return the deck’s contents similar to the output displayed below.
* A toString method for the hand to display the contents of the hand after the hand is ordered.

*Deck toString Output:*

*King of Spades*

*Queen of Spades*

*Jack of Spades*

*10 of Spades*

*9 of Spades*

*…*

*Ace of Spades*

*King of Hearts*

*…*

*Ace of Hearts*

*King of Diamonds*

*…*

*Ace of Diamons*