```
using System;

namespace CardGame
{
    // execute app
    class CardGameTest
    {
        public static void Main(string[] args)
        {
            Deck deck = new Deck();
            deck.shuffle();

            // display all 52 cards in the order they where dealt
            for(int i = 0; i < 52; i++)
            {
                Console.Write("{0, -19}", deck.DealCard());

                // create a new row after four columns
                if((i + 1) % 4 == 0)
                {
                    Console.WriteLine();
                }
            }
            Console.ReadKey();
        } // end Main
    } // end CardGameTest
}
```

```
// class Card represents a playing card
namespace CardGame
{
    public class Card
    {
        private string face;
        private string suit;

        // cconstructor initializes card's face and suit
        public Card(string cardFace, string cardSuit)
        {
            face = cardFace; // initialize face of card
            suit = cardSuit; // initialize suit of card
        }

        // override inherited ToString() method and return string
        // representaion of Card
        public override string ToString()
        {
            return face + " of " + suit;
        }
    }
}
```

```
using System;

namespace CardGame
{
    // class Deck represents a deck of cards
    public class Deck
    {
        private Card[] deck; // array of card objects
        private int currentCard; // index of next card
        private const int NUMBER_OF_CARDS = 52; // number of cards
        private Random randomNumbers; // random number generator

        // constructor fills deck with cards
        public Deck()
        {
            string[] faces = { "Ace", "Deuce", "Three", "Four", "Five", "Six",
                "Seven", "Eight", "Nine", "Ten", "Jack", "Queen", "King" };
            string[] suits = { "Hearts", "Diamonds", "Clubs", "Spades" };

            // Create array of Card objects
            deck = new Card[NUMBER_OF_CARDS];
            currentCard = 0;
            randomNumbers = new Random();

            // populate deck with card objects
            for (int count = 0; count < deck.Length; count++)
            {
                deck[count] = new Card(faces[count % 13], suits[count / 13]);
            }
        } // end constructor

        // shuffle deck
        public void shuffle()
        {
            // reset current card
            currentCard = 0;

            // for each card pick another random card and swap them
            for(int first = 0; first < deck.Length; first++)
            {
                // select a random number between 0 and 51
                int second = randomNumbers.Next(NUMBER_OF_CARDS);
                // swap current card with randomly selected card
                Card temp = deck[first];
                deck[first] = deck[second];
                deck[second] = temp;
            }
        } // end shuffle()

        // deal one card
        public Card DealCard()
        {
            // determmine whether cards remaiin to be dealt
            if(currentCard < deck.Length)
            {
                return deck[currentCard++]; // return current card in array
            }
            else
            {
                return null; // indicates that all cards have been dealt
            }
        } // end DealCard
    } // end Deck
}
```