Deck of Cards :

Write a complete program that implements the functionality of a deck of cards. In writing your program, use the provided DeckDriver and Card classes shown below. Write your own Deck class so that it works in conjunction with the two given classes. Use anonymous objects where appropriate.

Deck class details:

Use an ArrayList to store Card objects.

Deck constructor:

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. Study the Card class definition below for details.

dealCard:

This method removes the highest-indexed card in the ArrayList and returns it. In general, a method should not do more than what it’s supposed to do. Thus, dealCard should not print anything.

toString:

This method returns the deck’s contents using the format shown in the output session. In particular, note that toString should insert a newline after every fifth card. Hint: In coming up with a return value, use a String local variable. As you generate card values and newlines, concatenate those items to your local variable using the += operator.

Write your code such that the following classes produce the output shown in the subsequent output.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* DeckDriver.java

\* <your name>

\*

\* This class tests the Deck class.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public class DeckDriver

{

public static void main(String[] args)

{

Deck deck = new Deck();

System.out.println(deck.dealCard());

System.out.println(deck.dealCard());

System.out.println();

System.out.println(deck);

} // end main

} // end DeckDriver class

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Card.java

\* <your name>

\*

\* This class stores a Card's information.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public class Card

{

private int num; // hold a number between 1 and 13

private char suit; // holds 'C' for clubs, 'D' for diamonds,

// 'H' for hearts, 'S' for spades

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public Card(int num, char suit)

{

this.num = num;

this.suit = suit;

} // end Card constructor

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Return the card's value in the form of a concatenated

// number and character.

// For example, 1C = ace of clubs, 12H = queen of hearts.

public String toString()

{

return Integer.toString(num) + suit;

}

} // end class Card

Output:

13S

12S

1C 2C 3C 4C 5C

6C 7C 8C 9C 10C

11C 12C 13C 1D 2D

3D 4D 5D 6D 7D

8D 9D 10D 11D 12D

13D 1H 2H 3H 4H

5H 6H 7H 8H 9H

10H 11H 12H 13H 1S

2S 3S 4S 5S 6S

7S 8S 9S 10S 11S