|  |  |
| --- | --- |
| **Exposure Java** | **Lab 11b** |
| **The <Deck> Class Program** | **80 & 100 Point Versions** |
| **Assignment Purpose:**  This assignment is meant to demonstrate how to use a static one-dimensional array in a class, including assigning values and displaying the array. | |

Chapter IX introduced the **Card** class. This class was used to explain encapsulation. The **Card** class is also a fundamental class in the *Elevens AP® Lab*. Nothing was mentioned about the Card Game program that is a form of solitaire, called *Elevens.* In this chapter the previously introduced **Card** class, is now used to create a **Deck** class. In a future chapter both the **Card** class and the **Deck** class will be used in the Elevens solitaire card game.

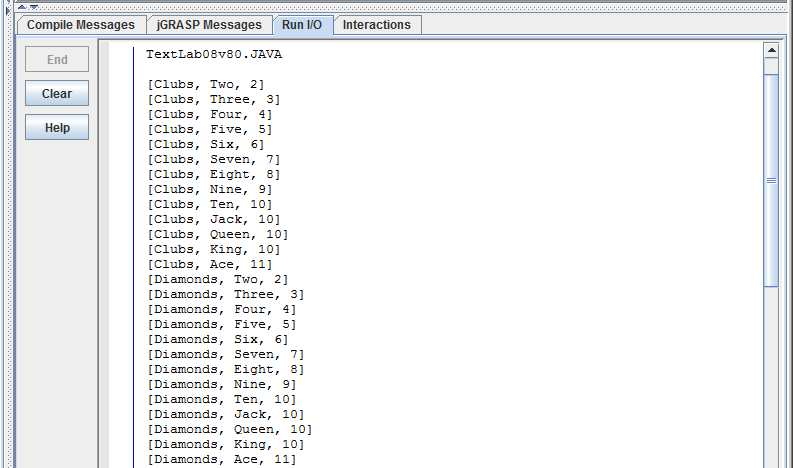
Chapter XI showed how the **cards** array is used as an attribute in the **Deck** class to store **Card** objects. This lab assignment is meant to improve the **Deck** class. The Student Starting version, shown below, shows a minimal **Deck** class.

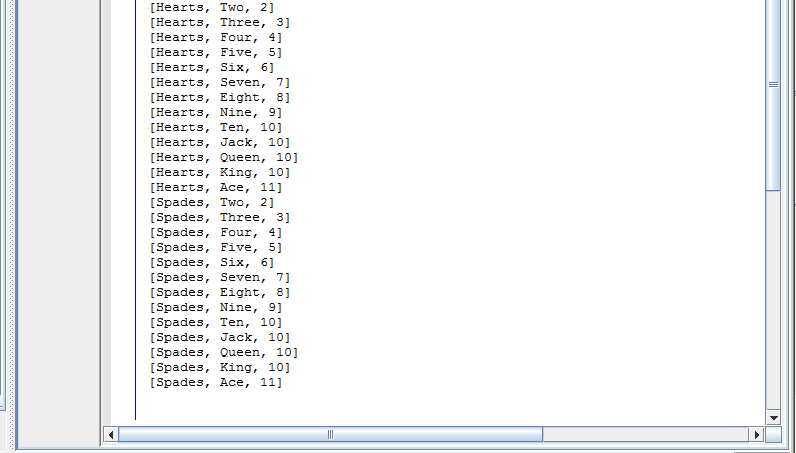
|  |  |
| --- | --- |
| **Lab11bvst Student Version** | **Do not copy this file, which is provided.** |
| // Lab11bvst.java // This is the Student starting version for the <Deck> class lab 11b assignment.    public class Lab11bvst {  public static void main(String[] args)  {  Deck deck = new Deck();  System.out.println(deck);  } }   class Deck  {  private Card[] cards;  private int size;   public Deck()   {  size = 52;  cards = new Card[size];  }   } | |

**80 Point Version Specifics**

For the 80-point version you need to rewrite the constructor so that all 52 cards of a normal card deck are assigned to the **cards** array. Keep in mind that card information needs to be stored inside the **Deck** class and is not passed by parameter. Additionally, you need to re-define the **toString** method for the **Deck** class so that it can be used to display the attribute values in a convenient manner. Make sure to take advantage of the **toString** method that already exists in the **Card** class.

**80 Point Version Output**





**Hint:**

You can store attributes in the **Deck** class that have an initializer list as shown below:

**private String[ ] suits = {"Clubs","Diamonds","Hearts","Spades"};**

**100 Point Version Specifics**

For the 100-point version you need to add a **shuffle** method, which is called from the constructor. The **shuffle** method is a **private** *helper method* in the **Deck** class. For this version you need to *shuffle* the deck by swapping the cards. Generate two random numbers in the **[0..51]** number range that will represent the indexes of the **cards** array and swap the cards. Make 1000 swaps and then display the cards. Use **Math.random** to generate random numbers.

**100 Point Version Partial Output**

