Project 4 - A Card Class

Due Date: April 7, 2009 @ 11:55 p.m.

You have been given a **skeleton** for a class called Card (**Card.java**) which represents a playing card in a standard deck of Cards. For the first part of the assignment, you will fill out the code so that the Card class works as required. Each Card object has two integer attributes - suit and rank. The suit is a number between 1 and 4 and the rank is a number between 2 and 14. The mapping between suit numbers and suits is as follows:

Suit Code	Suit
1	Club
2	Diamond
3	Hearts
4	Spade

The mapping between rank numbers and ranks is as follows:

Rank Number	Rank
2-10	Same
11	Jack
12	Queen
13	King
14	Ace

The Card class should allow any card in a standard deck of playing cards to be created, but it should NOT allow any other numbers for the suit or the rank. If someone tries to create a card with a suit of 15 and/or a rank of 100, for example, the program should default to some card that is actually in the deck (the ace of spades, for example). You will need to fill out each method as described in the comments at the top of the method. As you are writing the code, use the **CardTester** program to test your class to make sure it is working correctly. When every method of your Card class is properly implemented with no errors, the output will look something like this:

```
Rank of Card 1 is correct!
Rank of Card 2 is correct!
```

```
Suit of Card 1 is correct!
Suit of Card 2 is correct!
Card 1: 2 of Hearts
toString is working correctly for card 1
Card 2: A of Spades
toString is working correctly for card 2
Randomly Generated card: J of Clubs
Invalid Card defaulted to: A of Spades
The A of Spades was correctly identified as being greater than
the 2 of Hearts
```

Additionally, write a client class called **RandomCard.java** that randomly generates 15 Card objects (you can just use the default constructor) and prints them out, one by one. At the end of printing out the 15 cards, your program should print the highest card generated, the lowest card generated, and the number of times that each suit appeared among the cards The output of your program may look similar to the following:

15 randomly generated cards:

```
5 of Hearts
3 of Diamonds
J of Clubs
9 of Diamonds
6 of Spades
K of Hearts
2 of Clubs
7 of Clubs
Q of Diamonds
4 of Spades
2 of Clubs
3 of Hearts
A of Spades
10 of Diamonds
4 of Hearts

Highest card: A of Spades
Lowest card: 2 of Clubs

Number of Clubs: 4
Number of Diamonds: 4
Number of Hearts: 4
Number of Spades: 3
Number of Aces:1
```

Deliverables

- Card.java
- RandomCard.java

Hints

Card.java

- **Default Constructor**: use the method random of the Math class or the object of a Random class to generate a random number between 2 and14 for rank number, and a random number between 1 and 4 for suit code.

RandomCard.java

- Use a **for** loop and one Card object reference variable to create the 15 random card objects.
- In each iteration of the loop:
  - Use an assignment statement to create a new Card object and reassign it to the Card object reference variable
  - Keep track of the number of times each suit appeared among the cards.
  - Determine the highest and lower card.
- You need one counter variable for each suit.
- Use the method **compareTo** to obtain the lowest and highest card.
  - Create an object minCard that has the highest suit and rank using the appropriate constructor of the class Card
  - Create an object maxCard that has the lowest suit and rank using the appropriate constructor of the class Card
  - Within the for loop:
    - Use the method **compareTo** to check whether a new generated card is higher than the current maxCard
    - Use the method **compareTo** to check whether the current minCard is higher than the new generated Card.