

**Project 4 (100 points)****Due 11/15/2022**

Submit your solutions to canvas. For programming assignments do not send the entire project. All I want are the files ending in .java. **Each class will have to be defined in its own separate .java file.** All driver code should be put in one .java file. **Please make sure your name is included** at the top of each .java file. **There will be no re-submission of work. There will be no exceptions. So, before you submit your work please ensure you have everything the way you want it.**

**Problem1**

Copy the `Card` class from lectures 16 and 17 into your project folder and make sure it compiles. From there you are to implement a `High-Low Card` game. The first class you are going to make is a class called `Deck`. The data members for deck might be:

```
public static final int NCARDS = 52;
private Card[]        deck;
private int            cardsUsed;
private final int      SHUFFLENUM = 1000;
private RandomInteger  r1;
private RandomInteger  r2;
```

Some methods that might be in the `Deck` class are:

Constructors		
Constructor	Description	
<code>Deck ()</code>	Constructor for objects of class Deck.	

Method Summary		
All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
int	<code>CardsLeft ()</code>	Method to get number of cards left in the deck.
Card	<code>DealACard ()</code>	Method to deal one card from the deck and return it.
void	<code>Shuffle ()</code>	Method to put all used cards back into the deck and shuffle them into a random order.

The next class you will need to create is the HighLow class. The data members are:

```
private int    m_gamesPlayed;
private int    m_sumOfScores;
```

Some important methods are:

[illegible]

```

////////////////////////////////////
//
// Displays the current card that the user will base the guess of high or low from.
//
// parameters: card of type Card representing the last card dealt from the deck. The user
//              will guess if the next card dealt is higher or lower than this.
// return: none
//
////////////////////////////////////
protected void DisplayCurrentCard(Card card)

////////////////////////////////////
//
// Displays the next card dealt from the deck after the user's guess. This card will
// be compared to see if it is higher or lower than the last card dealt.
//
// parameters: card of type Card representing the last card dealt from the deck.
//              This card will be compared to the last card dealt, along with the
//              user's guess to determine a right or wrong guess.
//
////////////////////////////////////
protected void DisplayNextCard(Card card)

////////////////////////////////////
//
// Display result of a round of High/Low
//
// parameters: result of type String indicating whether the user won or lost based
//              on the last card dealt, and the player's guess.
//
// return: None
//
////////////////////////////////////
protected void DisplayResult(String result)

////////////////////////////////////
//
// Displays the finals stats of all the rounds of Hi/Low played (displays the
// average score) It computes the average score using the data members
// gamesPlayed, and sumOfScores.
//
// parameters: none.
// return type none.
//
////////////////////////////////////
protected void DisplayFinalStats()

```

Below is some sample output from the game:

```

The current card is Queen of Spades
Will the next card be higher (H) or lower (L)?
L
The next card is 5 of Spades
Your prediction was correct.
The current card is 5 of Spades
Will the next card be higher (H) or lower (L)?

```

H

The next card is 8 of Diamonds

Your prediction was correct.

The current card is 8 of Diamonds

Will the next card be higher (H) or lower (L)?

H

The next card is Ace of Clubs

Your prediction was incorrect.

The game is over.

You made 2 correct predictions.

Play again?

Y

The current card is 7 of Clubs

Will the next card be higher (H) or lower (L)?

H

The next card is King of Diamonds

Your prediction was correct.

The current card is King of Diamonds

Will the next card be higher (H) or lower (L)?

L

The next card is 9 of Clubs

Your prediction was correct.

The current card is 9 of Clubs

Will the next card be higher (H) or lower (L)?

L

The next card is 3 of Hearts

Your prediction was correct.

The current card is 3 of Hearts

Will the next card be higher (H) or lower (L)?

H

The next card is 6 of Hearts

Your prediction was correct.

The current card is 6 of Hearts

Will the next card be higher (H) or lower (L)?

H

The next card is 9 of Spades

Your prediction was correct.

The current card is 9 of Spades

Will the next card be higher (H) or lower (L)?

L

The next card is 5 of Hearts

Your prediction was correct.

The current card is 5 of Hearts

Will the next card be higher (H) or lower (L)?

H

The next card is 10 of Diamonds  
Your prediction was correct.  
The current card is 10 of Diamonds  
Will the next card be higher (H) or lower (L)?  
L  
The next card is 3 of Spades  
Your prediction was correct.  
The current card is 3 of Spades  
Will the next card be higher (H) or lower (L)?  
H  
The next card is Queen of Diamonds  
Your prediction was correct.  
The current card is Queen of Diamonds  
Will the next card be higher (H) or lower (L)?  
L  
The next card is 9 of Hearts  
Your prediction was correct.  
The current card is 9 of Hearts  
Will the next card be higher (H) or lower (L)?  
L  
The next card is 2 of Spades  
Your prediction was correct.  
The current card is 2 of Spades  
Will the next card be higher (H) or lower (L)?  
H  
The next card is 3 of Clubs  
Your prediction was correct.  
The current card is 3 of Clubs  
Will the next card be higher (H) or lower (L)?  
H  
The next card is 8 of Spades  
Your prediction was correct.  
The current card is 8 of Spades  
Will the next card be higher (H) or lower (L)?  
L  
The next card is 2 of Clubs  
Your prediction was correct.  
The current card is 2 of Clubs  
Will the next card be higher (H) or lower (L)?  
H  
The next card is King of Clubs  
Your prediction was correct.  
The current card is King of Clubs  
Will the next card be higher (H) or lower (L)?  
L  
The next card is Queen of Hearts  
Your prediction was correct.

```
The current card is Queen of Hearts
Will the next card be higher (H) or lower (L)?
L
The next card is Queen of Spades
You lose on ties.  Sorry!

The game is over.
You made 16 correct predictions.

Play again?
n
Average score of 9.0 for 2 rounds played.
```

The last part of this project is going consist of turning the text display into graphics for the cards and a GUI for the game control. You can customize `GPairOfDice.java` from the lect13 module. You will create a `GHighLow` class that extends `HighLow` and overrides the following methods:

```
@Override
protected int PlayARound()

@Override
public void DisplayCurrentCard(Card card)
{
    SetCardLabel(card, this.iconLabelOne);
}

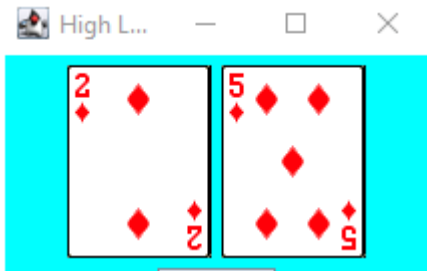
@Override
public void DisplayNextCard(Card card)
{
    SetCardLabel(card, this.iconLabelTwo);
}
```

Below is a mixture of graphics for the cards and text displays for the GUI. The ultimate goal is to completely remove the text GUI.



Will the next card be higher (H) or lower (L)?

H



Your prediction was correct.

Any key to continue

c



Will the next card be higher (H) or lower (L)?

H



Your prediction was correct.

Any key to continue

c



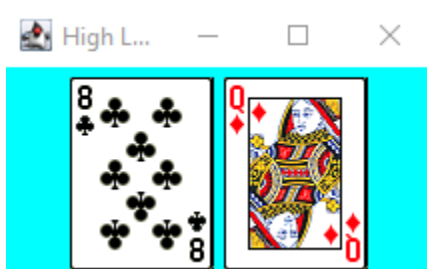
Will the next card be higher (H) or lower (L)?  
L



Your prediction was correct.  
Any key to continue  
C



Will the next card be higher (H) or lower (L)?  
L



Your prediction was incorrect.  
Any key to continue



c

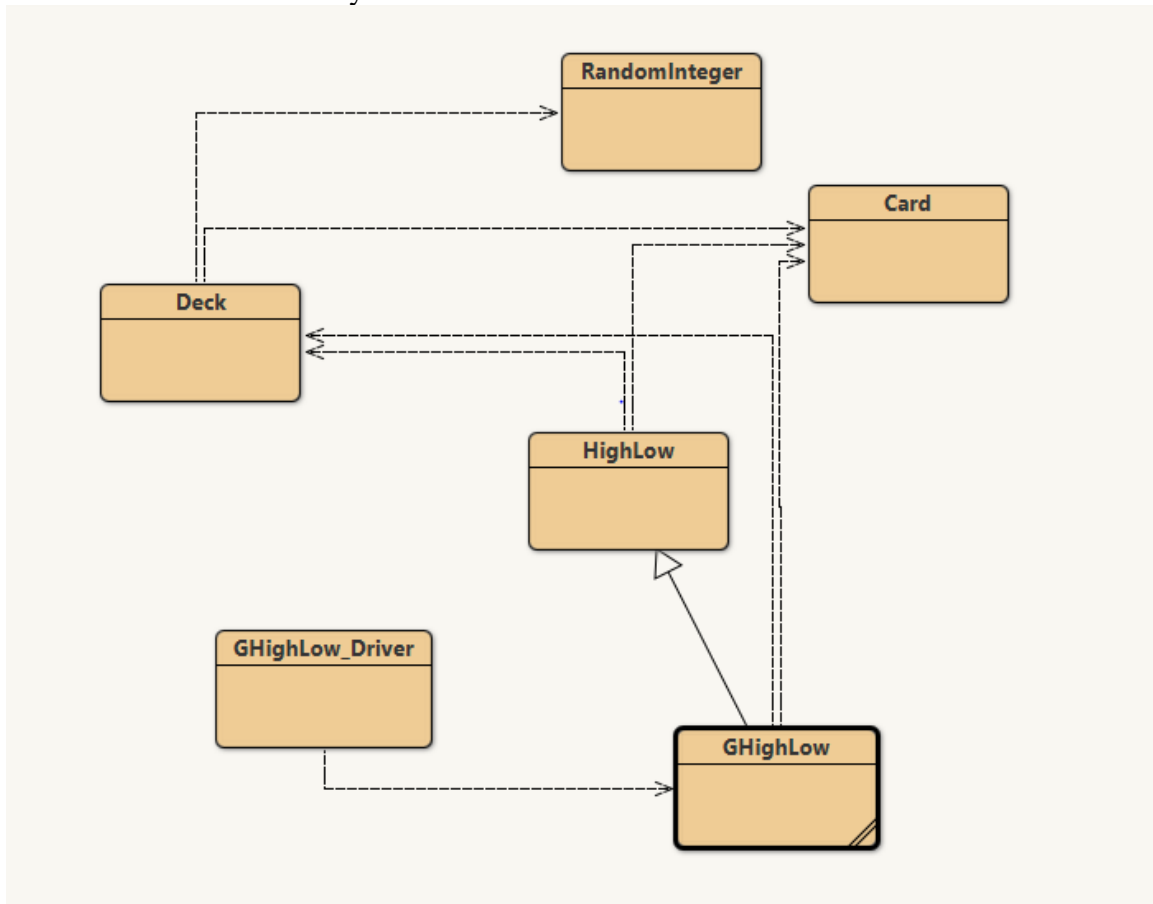
The game is over.  
You made 3 correct predictions.

Play again?

n

Average score of 3.0 for 1 rounds played.

Below is the class hierarchy:



A smart approach to this project is to build a little, test a little, build a little, test a little. For example, the first thing to do is create the `Deck` class. Code this up one method at a time. First create the constructor, then test. Then create `Shuffle`, then test. Then create the method `DealACard`, and then test. Finally implement the `CardsLeft` method.