

Programming 1 Project.

Important information:

1. You are required to have a working solution of Assignment 5 before starting this project.
2. This project will be divided into 3 parts. There won't be any submission link for Part 1 and 2 of this project, submission will be in class. You will run your code in class and walk me through how you came about it.
3. Absence from any class without showing me a complete solution of any project part that has been assigned, and without any notification will result in an automatic grade of 0 for that part.
4. Part 1 and Part 2 of this project are individual work. Part 3 is group work. Only students with a working solution of part 1 and part 2 will be allowed to participate in part 3. Hence a 0 in part 1 or part 2 will be an automatic 0 for the entire project.

Project Part 1:

In continuation from the classes created in Assignment 5, Card.java, Joker.java, Deck.java, and Game.java (classes, variables and methods must be named exactly as instructed), add or edit the following classes:

Player.java

The Player class is responsible for representing a Player. It has the following variables:

```
private String name;  
private Deck playerDeck;
```

and the following methods:

A constructor: that takes one parameter, name. Inside the constructor, add the following line that initializes the deck variable:

```
playerDeck = new Deck(54);
```

(A second constructor will be created in the Deck class)

Getters for the class variables.

Deck.java

Add the following variable to your Deck class in addition to what is already there:

```
private int numCards;
```

Add the following methods to your Deck class in addition to what is already there:

A parameterized constructor: This constructor takes an int parameter that represents the deck size, and initializes the Card[] deck variable as follows

```
deck = new Card[size(or whatever you call your int variable)];
```

Set the numCards variable to 0.

This is all it does, it doesn't create 54 cards like the default constructor.

public void addToDeck(Card card): This method will be responsible for adding a card to the deck at numCards index, and increment numCards. (ensure numCards is within the range of the size of the deck).

public void printDeck(): This method simply prints all the cards in the deck.

Driver class

You can name this class anything you want to name it.

In the main method, ask the user for 2 names, and create 2 players with the 2 names.

Create a standard deck of cards using the default constructor that takes no parameters.

Split the deck between the two players and print each player's deck.

(Hint, you will need a loop that runs 27 times. Each time through the loop, take 2 cards from the standard deck, one for each player, and add to each of the player's decks using the addToDeck method.)