

Programming 1 Assignment 5.

Submit a single .txt file containing your solution(all your java files) to Lea.

Question 1:

This assignment serves as an intro to the project. The overall aim of the project is to create the popular Card game, War.

However for this assignment, we will start creating the various classes we will need and implement just a simple guessing game to ensure our classes are working as they should.

You are required to create 3 classes (must be named exactly the same else points will be deducted): Deck.java, Card.java, Game.java.

Card.java

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

```
private int suit; (spades, hearts, diamonds, clubs or joker represented numerically)
private int value; (the numerical values on each card, 1 to 13, except joker which is only 2 in number)
```

and the following methods:

A constructor: that takes two parameters, one for the value and one for the suit, then sets it to the class variables.

Getters for the class variables.

public String getSuitAsString(): As the name implies, this method will use a switch case statement to return string representation of a card's suit based on the suit numerical value where 1 = Spades, 2 = Hearts, 3 = Diamonds, 4 = Clubs, 5 = Joker.

public String getValueAsString(): As the name implies, this method will use a switch case statement to return string representation of a card's value based on the numerical value where 1 = Ace, 2 to 10 simply returns the numbers in quotation marks, 11 = Jack, 12 = Queen, 13 = King. But if the card's suit is a Joker, then simply return whatever the value is in quotation marks, because there is no Ace of joker.

public String toString(): This will return a string representation of a Card by calling the getValueAsString() method and getSuitAsString() method, space separated with

“of” between them, eg “Queen of Hearts”, “10 of Diamonds”, “Ace of Spades”, “1 of Joker” etc

Deck.java

The Deck class is responsible for representing and constructing a deck of 54 cards. It will have the following variables:

`private Card[] deck;` (An array of 54 cards: 52 regular cards and 2 jokers)

and the following methods:

A constructor: that takes no parameters and creates 54 cards.

To do this, the constructor will initialize the deck array variable with 54 elements.

Then using two nested for loops, the outer loop representing the suit of the card(there are 4 suits), the inner loop representing the numerical value of the card(there are 14 of them), and using an extra variable to keep track of index of the array from 0 to 53, 52 cards will be created in the nested for loop and stored in the deck array.

When the loop has been exited, the two joker cards will be created and saved in index 52 and 53. They can be given suite 5 and numbers 1 and 2.

public void shuffle(): This will be responsible for shuffling the deck(mixing them up in random order). Copy the code below into your class:

```
public void shuffle() {
    for ( int i = deck.length-1; i > 0; i-- ) {
        int rand = (int) (Math.random() * (i+1));
        Card temp = deck[i];
        deck[i] = deck[rand];
        deck[rand] = temp;
    }
}
```

public Card dealCard(int index): This method will be responsible for returning a card from the deck at the index sent into the method.

Game.java

This is the driver class that ties everything together in a main method. In the main method, create a deck of cards by creating an object instance of the Deck class. Shuffle the deck.

Using a scanner object, ask the user to guess a card (which they will type in as a string in the form of “ace of diamonds” etc but they can’t guess joker).

Then ask them to pick a card from the deck(they will do this by entering an int between 0 – 53), tell them what card they picked by printing the card.

If the card they guessed and the card they picked are the same, or they picked a joker, then they win and the game ends.

If not, they have 5 tries (shuffle the deck after each guess), after which the game ends if no match and they loose.