Intro to Java Week 6 Coding Assignment

Points possible: 70

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

Instructions: In Eclipse, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your Java project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Coding Steps:

For the final project you will be creating an automated version of the classic card game *WAR*.

1. Create the following classes.
    a. Card
        i. Fields
            1. **value** (contains a value from 2-14 representing cards 2-Ace)
            2. **name** (e.g. Ace of Diamonds, or Two of Hearts)
        ii. Methods
            1. Getters and Setters
            2. **describe** (prints out information about a card)
    b. Deck
        i. Fields
            1. **cards** (List of Card)
        ii. Methods
            1. **shuffle** (randomizes the order of the cards)
            2. **draw** (removes and returns the top card of the Cards field)

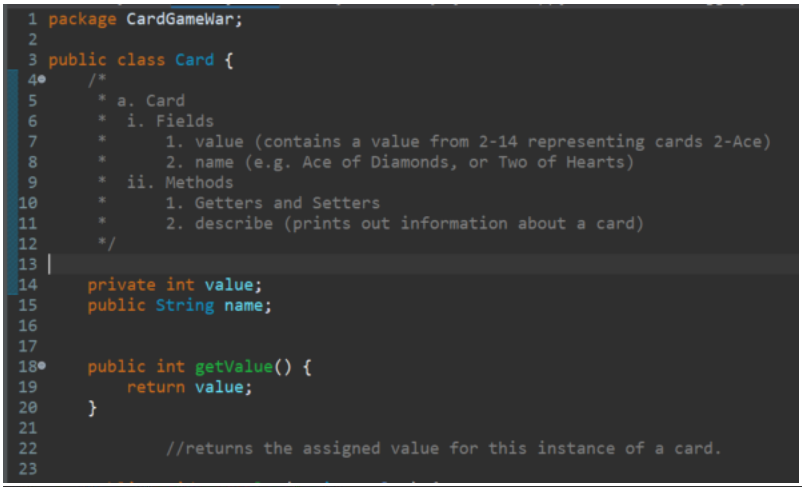
3. In the constructor, when a new Deck is instantiated, the Cards field should be populated with the standard 52 cards.

c. Player
   i. Fields
      1. **hand** (List of Card)
      2. **score** (set to 0 in the constructor)
      3. **name**
   ii. Methods
      1. **describe** (prints out information about the player and calls the describe method for each card in the Hand List)
      2. **flip** (removes and returns the top card of the Hand)
      3. **draw** (takes a Deck as an argument and calls the draw method on the deck, adding the returned Card to the hand field)
      4. **incrementScore** (adds 1 to the Player's score field)

2. Create a class called App with a main method.
3. Instantiate a Deck and two Players, call the shuffle method on the deck.
4. Using a traditional for loop, iterate 52 times calling the Draw method on the other player each iteration using the Deck you instantiated.
5. Using a traditional for loop, iterate 26 times and call the flip method for each player.
   a. Compare the value of each card returned by the two player's flip methods. Call the incrementScore method on the player whose card has the higher value.
6. After the loop, compare the final score from each player.
7. Print the final score of each player and either "Player 1", "Player 2", or "Draw" depending on which score is higher or if they are both the same.

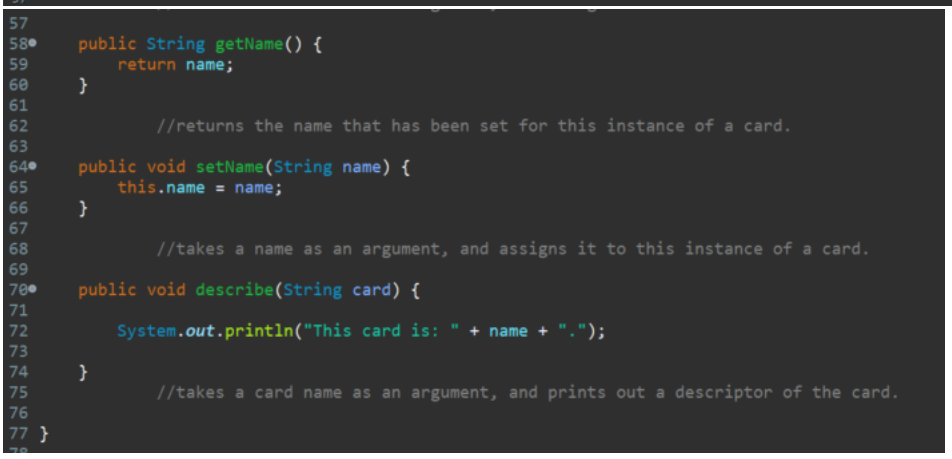
Screenshots of Code:

Card class code:



```java
package CardGameWar;

public class Card {
    /*
     * a. Card
     *  i. Fields
     *      1. value (contains a value from 2-14 representing cards 2-Ace)
     *      2. name (e.g. Ace of Diamonds, or Two of Hearts)
     *  ii. Methods
     *      1. Getters and Setters
     *      2. describe (prints out information about a card)
     */

    private int value;
    public String name;


    public int getValue() {
        return value;
    }

            //returns the assigned value for this instance of a card.
```

```java
    public void setValue(String value) {

        if(value.contains("Ace")) {
            this.value = 2;
        } else if(value.contains("Two")) {
            this.value = 3;
        } else if(value.contains("Three")) {
            this.value = 4;
        } else if(value.contains("Four")) {
            this.value = 5;
        } else if(value.contains("Five")) {
            this.value = 6;
        } else if(value.contains("Six")) {
            this.value = 7;
        } else if(value.contains("Seven")) {
            this.value = 8;
        } else if(value.contains("Eight")) {
            this.value = 9;
        } else if(value.contains("Nine")) {
            this.value = 10;
        } else if(value.contains("Ten")) {
            this.value = 11;
        } else if(value.contains("Jack")) {
            this.value = 12;
        } else if(value.contains("Queen")) {
            this.value = 13;
        } else if(value.contains("King")) {
            this.value = 14;
        }

    }

            //takes a card name as an argument, and assigns a value to this instance of a card based on what card to be used for comparison on who won a round.
```



```java

    public String getName() {
        return name;
    }

            //returns the name that has been set for this instance of a card.

    public void setName(String name) {
        this.name = name;
    }

            //takes a name as an argument, and assigns it to this instance of a card.

    public void describe(String card) {

        System.out.println("This card is: " + name + ".");

    }
            //takes a card name as an argument, and prints out a descriptor of the card.

}
```

Deck class code:

```java
package CardGameWar;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.List;

public class Deck {

    //b.    Deck
    /*
     * i. Fields
     *      1. cards (List of Card)
     *
     * ii. Methods
     *      1. shuffle (randomizes the order of the cards)
     *      2. draw (removes and returns the top card of the Card field)
     */

    public String[] deckOfCards = {"Ace of Spades","Two of Spades","Three of Spades","Four of Spades","Five of Spades","Six of Spades","Seven of Spades","Eight of Spades","Nine of Spades","Ten of Spades",
            "Ace of Clubs","Two of Clubs","Three of Clubs","Four of Clubs","Five of Clubs","Six of Clubs","Seven of Clubs","Eight of Clubs","Nine of Clubs","Ten of Clubs","Jack of Clubs","Queen of Clubs",
            "Ace of Diamonds","Two of Diamonds","Three of Diamonds","Four of Diamonds","Five of Diamonds","Six of Diamonds","Seven of Diamonds","Eight of Diamonds","Nine of Diamonds","Ten of Diamonds","Ja
            "Ace of Hearts","Two of Hearts","Three of Hearts","Four of Hearts","Five of Hearts","Six of Hearts","Seven of Hearts","Eight of Hearts","Nine of Hearts","Ten of Hearts","Jack of Hearts","Queer

        //String array that holds every card that can be found in a standard deck.

    List<String> deck = new ArrayList<> (Arrays.asList(deckOfCards));
            //converts deckOfCards into a list to be used more easily in other methods.


    public List<String> shuffle() {

        Collections.shuffle(deck); //shuffles the deck.

        return deck; //returns the shuffled deck.
    }

    public String draw(List<String> deckInPlay) {

        String drawnCard = deckInPlay.get(0);

        deckInPlay.remove(0);

        return drawnCard;

        //takes a deck as an argument, pulls and stores the top card, removes that top card from the deck, and outputs the card that was drawn.
    }
```

```java
    public String draw(List<String> deckInPlay) {

        String drawnCard = deckInPlay.get(0);

        deckInPlay.remove(0);

        return drawnCard;

        //takes a deck as an argument, pulls and stores the top card, removes that top card from the deck, and outputs the card that was drawn.
    }


    public List<String> getDeckOfCards() {

        return deck; //returns the deck as a list for used as an argument in the draw method.

    }

}
```

Player class code:

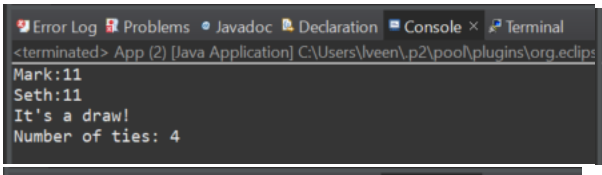
```java
package CardGameWar;

import java.util.ArrayList;
import java.util.List;

public class Player {
    /*
     * c. Player
     *  i. Fields
     *      1. hand (List of Card)
     *      2. score (set to 0 in the constructor)
     *      3. name
     *  ii. Methods
     *      1. describe (prints out information about the player and calls the describe method for each card in the Hand List)
     *      2. flip (removes and returns the top card of the Hand)
     *      3. draw (takes a Deck as an argument and calls the draw method on the deck, adding the returned Card to the hand field)
     *      4. incrementScore (adds 1 to the Player's score field)
     */

    public List<String> hand = new ArrayList<String>();
    public int score = 0;
    private String name;

    Card playerCard = new Card();
    Deck playerDeck = new Deck();
        //instantiates a card and a deck to be used in later methods for a player.

    public void describe() {
        System.out.println("Player " + name + " has " + score + " points.");

        for(int i = 0; i < hand.size(); i ++) {
            playerCard.setName(hand.get(i));
            playerCard.describe(hand.get(i));

        }
    }

        //prints out the player name and score, then for every card in the hand it gives it a name which is then used in the describe method.
        //the describe method prints out a description of the card.

    public String flip() {

        String flippedCard = hand.get(0);
            //stores the top card as a string.
        hand.remove(hand.get(0));
            //removes the top card from the player's hand.
        return flippedCard;
            //returns the stored top card.
    }

    public void draw(List<String> deckInPlay) {

        hand.add(playerDeck.draw(deckInPlay));

            //takes the deck for the current game as an argument, calls the draw method on it, and adds that card to the player's hand.

    }

    public void incrementScore() {

        score ++;
            //adds one to the player's score when called.
    }

    public String getName() {
        return name;
            //returns the player's name when called.
    }

    public void setName(String name) {
        this.name = name;
            //takes a name as an argument, and sets this player's name to that name.
    }

    public int getScore() {
        return score;
            //returns the score of this player when called.
    }
}
```

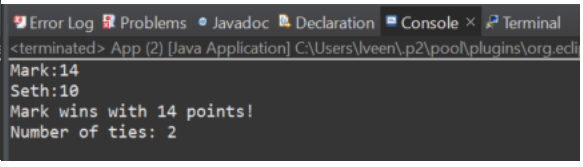
App class code:

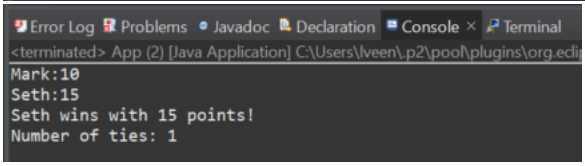
```java
package CardGameWar;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class App {

    public static void main(String[] args) {
        // TODO Auto-generated method stub


        Card playerOneCard = new Card();
        Card playerTwoCard = new Card();
            //creates a card instance for each player to be used in comparing which card is higher and award points.
        Deck gameDeck = new Deck();
        Player one = new Player();
        Player two = new Player();
            //creates a deck and two players for the game.

        int ties = 0;
            //creates a variable to hold the number of ties during play.
        one.setName("Mark");
        two.setName("Seth");
            //sets the player's names.

                    //System.out.println(gameDeck.getDeckOfCards());
                        //test code to see the deck.

        gameDeck.shuffle(); // shuffles all the cards in the named deck.

                    //System.out.println(gameDeck.getDeckOfCards());
                    //System.out.println(gameDeck.getDeckOfCards().size());
                        //test code to see if the deck shuffled correctly and how many cards there are in the deck.

        for (int i = 0; i < 52; i++) {
            if(i % 2 == 0) {
                one.draw(gameDeck.getDeckOfCards());
            } else {
                two.draw(gameDeck.getDeckOfCards());
            }
        }

                    //one.describe();
                    //two.describe();
                        //test code to check each player's hands for 26 cards.

```

```
47
48          for (int i = 0; i < 26; i++) {
49
50              playerOneCard.setValue(one.flip());
51              playerTwoCard.setValue(two.flip());
52                  //calls the flip method on each player and sets the value of the flipped card to each player's card.
53
54                      //System.out.println(playerOneCard.getValue());
55                      //System.out.println(playerTwoCard.getValue());
56                          //test code to check that values are assigned to player cards properly.
57
58              if(playerOneCard.getValue() > playerTwoCard.getValue()) {
59                  one.incrementScore(); //adds one to player one's score if their card had a higher value.
60              } else if (playerOneCard.getValue() < playerTwoCard.getValue()) {
61                  two.incrementScore(); //adds one to player two's score if their card had a higher value.
62              } else if (playerOneCard.getValue() == playerTwoCard.getValue()) {
63                  ties ++;    //adds one to ties if the player's card's values are equal.
64              }
65
66                  //all of this is run 26 times till both players have no more cards to flip.
67
68          }
69
70
71          System.out.println(one.getName() + ":" + one.getScore());
72          System.out.println(two.getName() + ":" + two.getScore());
73              //outputs the final scores.
74
75
76          if (one.getScore() > two.getScore()) {
77              System.out.println(one.getName() + " wins with " + one.getScore() + " points!");
78              System.out.println("Number of ties: " + ties);
79          } else if (two.getScore() > one.getScore()) {
80              System.out.println(two.getName() + " wins with " + two.getScore() + " points!");
81              System.out.println("Number of ties: " + ties);
82          } else if (one.getScore() == two.getScore()) {
83              System.out.println("It's a draw!");
84              System.out.println("Number of ties: " + ties);
85          }
86              //based on the final scores, outputs who won or if no one wins, as well as the number of ties.
87
88      }
89
90  }
```

Screenshots of Running Application:







Displays functionality when it is a draw, player 1(Mark) wins, and player 2(Seth) wins, as well as their scores and the number of ties.

URL to GitHub Repository:

https://github.com/LVeenendaal/Week-6-Project/upload/master/src/CardGameWar