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

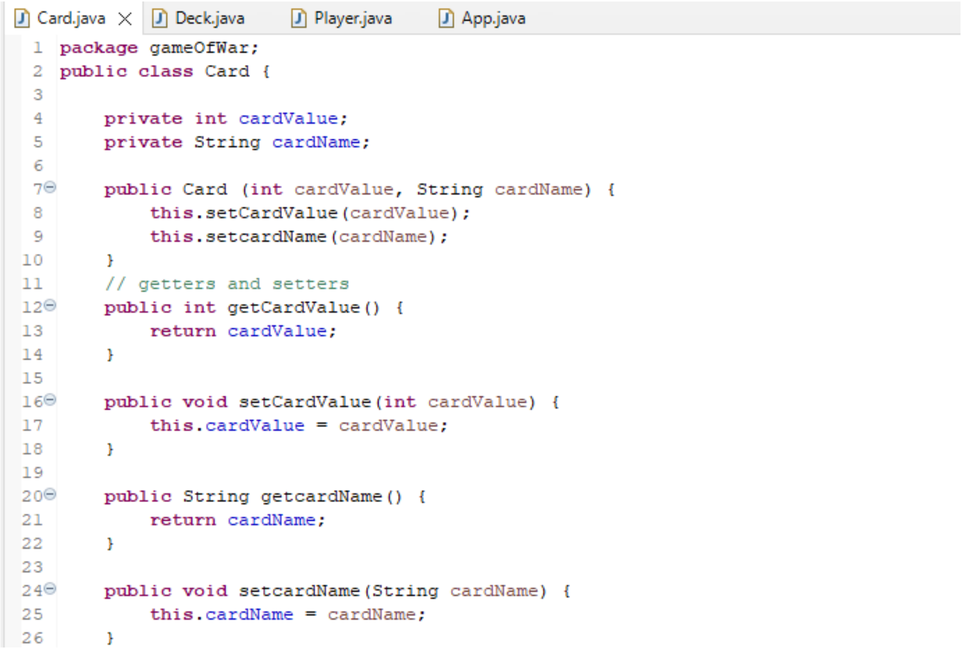
      ii. Methods

         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)

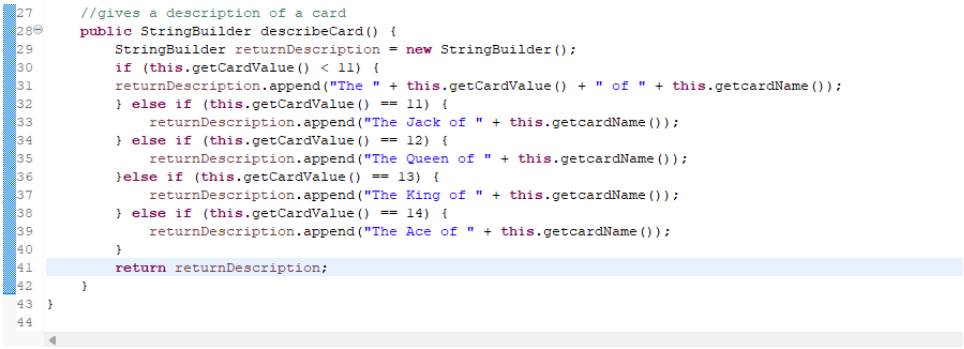
2. Create a class called App with a main method.
3. Instantiate a Deck and two Players, call the shuffle method on the deck.
4. Using a traditional for loop, iterate 52 times calling the Draw method on the other player each iteration using the Deck you instantiated.
5. Using a traditional for loop, iterate 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:

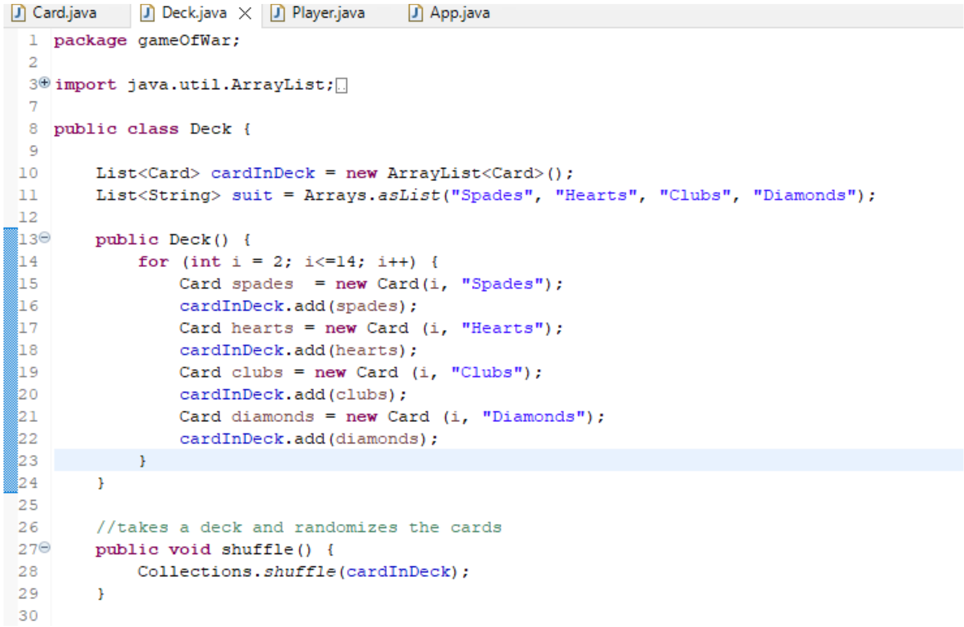


```
package gameOfWar;
public class Card {

    private int cardValue;
    private String cardName;

    public Card (int cardValue, String cardName) {
        this.setCardValue(cardValue);
        this.setcardName(cardName);
    }
    // getters and setters
    public int getCardValue() {
        return cardValue;
    }

    public void setCardValue(int cardValue) {
        this.cardValue = cardValue;
    }

    public String getcardName() {
        return cardName;
    }

    public void setcardName(String cardName) {
        this.cardName = cardName;
    }
```

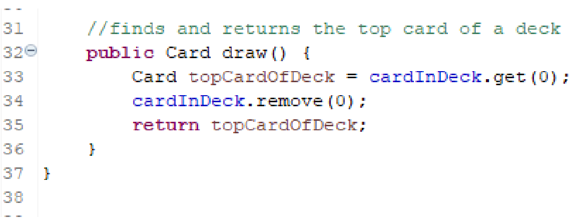


```
    //gives a description of a card
    public StringBuilder describeCard() {
        StringBuilder returnDescription = new StringBuilder();
        if (this.getCardValue() < 11) {
        returnDescription.append("The " + this.getCardValue() + " of " + this.getcardName());
        } else if (this.getCardValue() == 11) {
           returnDescription.append("The Jack of " + this.getcardName());
        } else if (this.getCardValue() == 12) {
           returnDescription.append("The Queen of " + this.getcardName());
        }else if (this.getCardValue() == 13) {
           returnDescription.append("The King of " + this.getcardName());
        } else if (this.getCardValue() == 14) {
           returnDescription.append("The Ace of " + this.getcardName());
        }
        return returnDescription;
    }
}

```

Deck Class:

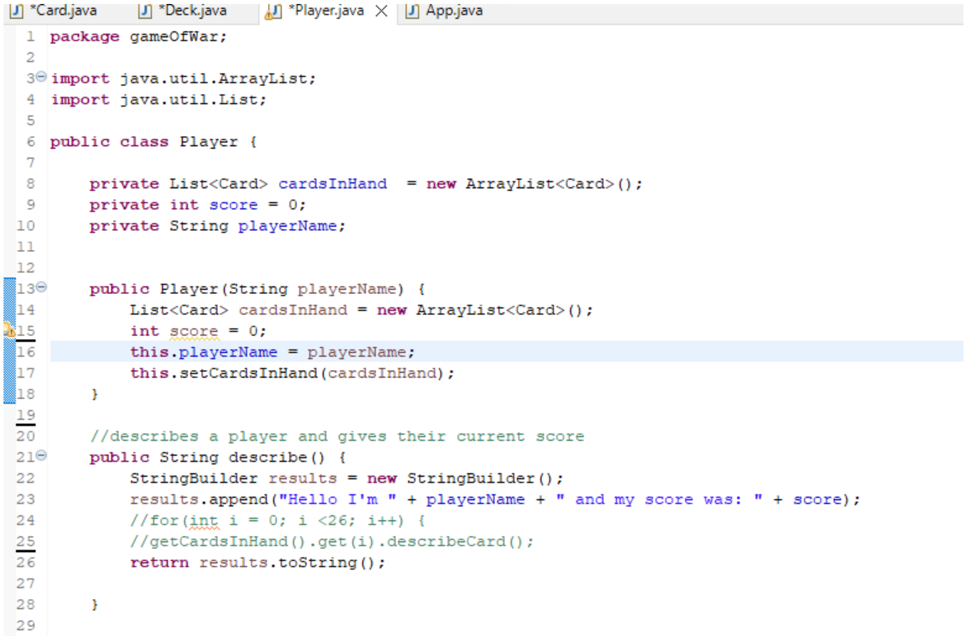


```
package gameOfWar;

import java.util.ArrayList;

public class Deck {

    List<Card> cardInDeck = new ArrayList<Card>();
    List<String> suit = Arrays.asList("Spades", "Hearts", "Clubs", "Diamonds");

    public Deck() {
        for (int i = 2; i<=14; i++) {
            Card spades  = new Card(i, "Spades");
            cardInDeck.add(spades);
            Card hearts = new Card (i, "Hearts");
            cardInDeck.add(hearts);
            Card clubs = new Card (i, "Clubs");
            cardInDeck.add(clubs);
            Card diamonds = new Card (i, "Diamonds");
            cardInDeck.add(diamonds);
        }
    }

    //takes a deck and randomizes the cards
    public void shuffle() {
        Collections.shuffle(cardInDeck);
    }

```

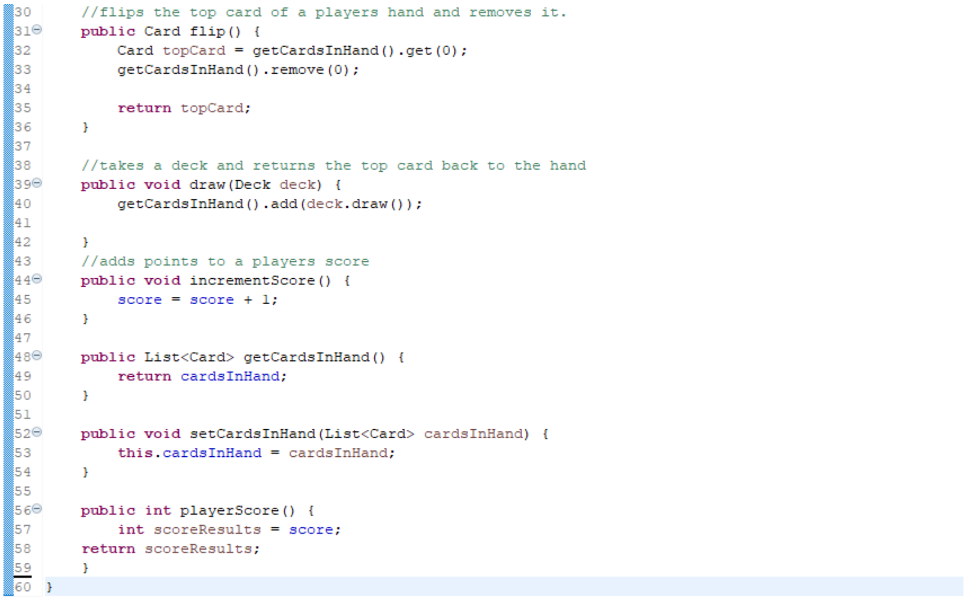


```
    //finds and returns the top card of a deck
    public Card draw() {
        Card topCardOfDeck = cardInDeck.get(0);
        cardInDeck.remove(0);
        return topCardOfDeck;
    }
}
```

Player Class:

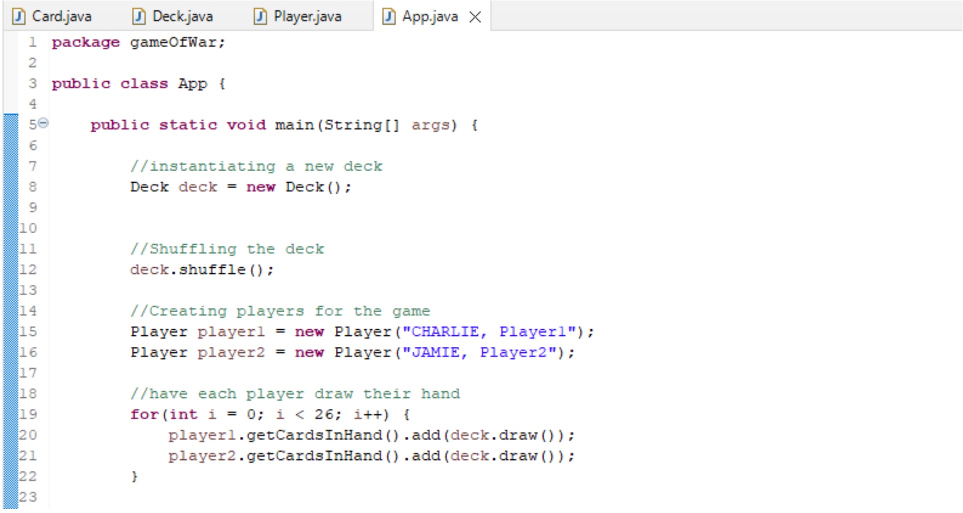


```
package gameOfWar;

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

public class Player {

    private List<Card> cardsInHand  = new ArrayList<Card>();
    private int score = 0;
    private String playerName;


    public Player(String playerName) {
        List<Card> cardsInHand = new ArrayList<Card>();
        int score = 0;
        this.playerName = playerName;
        this.setCardsInHand(cardsInHand);
    }

    //describes a player and gives their current score
    public String describe() {
        StringBuilder results = new StringBuilder();
        results.append("Hello I'm " + playerName + " and my score was: " + score);
        //for(int i = 0; i <26; i++) {
        //getCardsInHand().get(i).describeCard();
        return results.toString();

    }

```

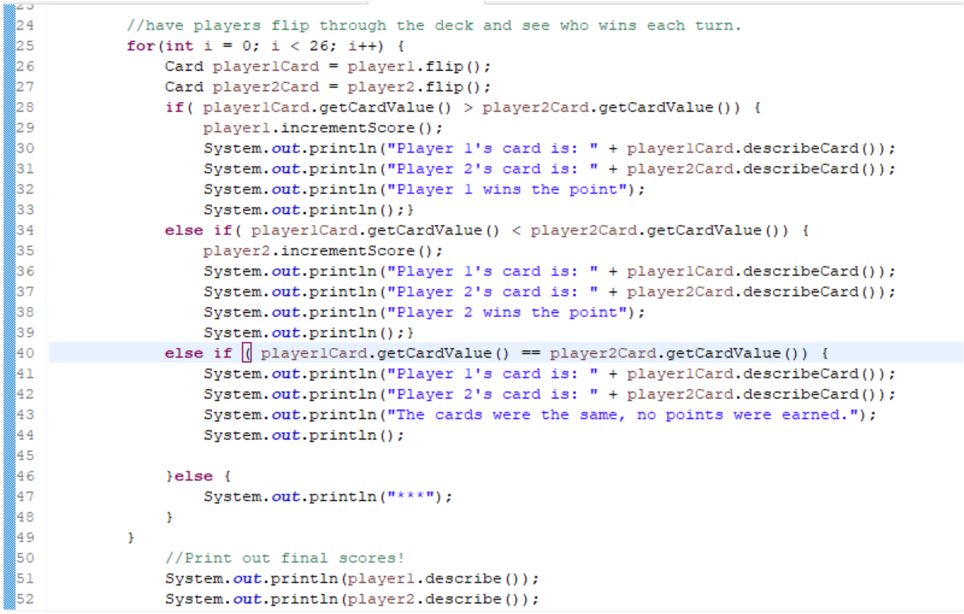


```
    //flips the top card of a players hand and removes it.
    public Card flip() {
        Card topCard = getCardsInHand().get(0);
        getCardsInHand().remove(0);

        return topCard;
    }

    //takes a deck and returns the top card back to the hand
    public void draw(Deck deck) {
        getCardsInHand().add(deck.draw());

    }
    //adds points to a players score
    public void incrementScore() {
        score = score + 1;
    }

    public List<Card> getCardsInHand() {
        return cardsInHand;
    }

    public void setCardsInHand(List<Card> cardsInHand) {
        this.cardsInHand = cardsInHand;
    }

    public int playerScore() {
        int scoreResults = score;
    return scoreResults;
    }
}
```

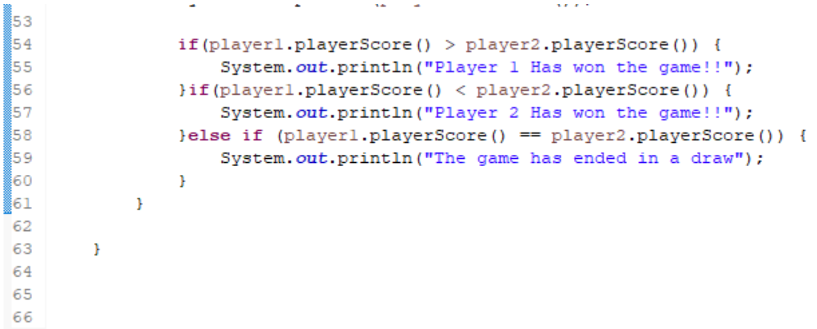
App Class:



```java
package gameOfWar;

public class App {

    public static void main(String[] args) {

        //instantiating a new deck
        Deck deck = new Deck();


        //Shuffling the deck
        deck.shuffle();

        //Creating players for the game
        Player player1 = new Player("CHARLIE, Player1");
        Player player2 = new Player("JAMIE, Player2");

        //have each player draw their hand
        for(int i = 0; i < 26; i++) {
            player1.getCardsInHand().add(deck.draw());
            player2.getCardsInHand().add(deck.draw());
        }

```

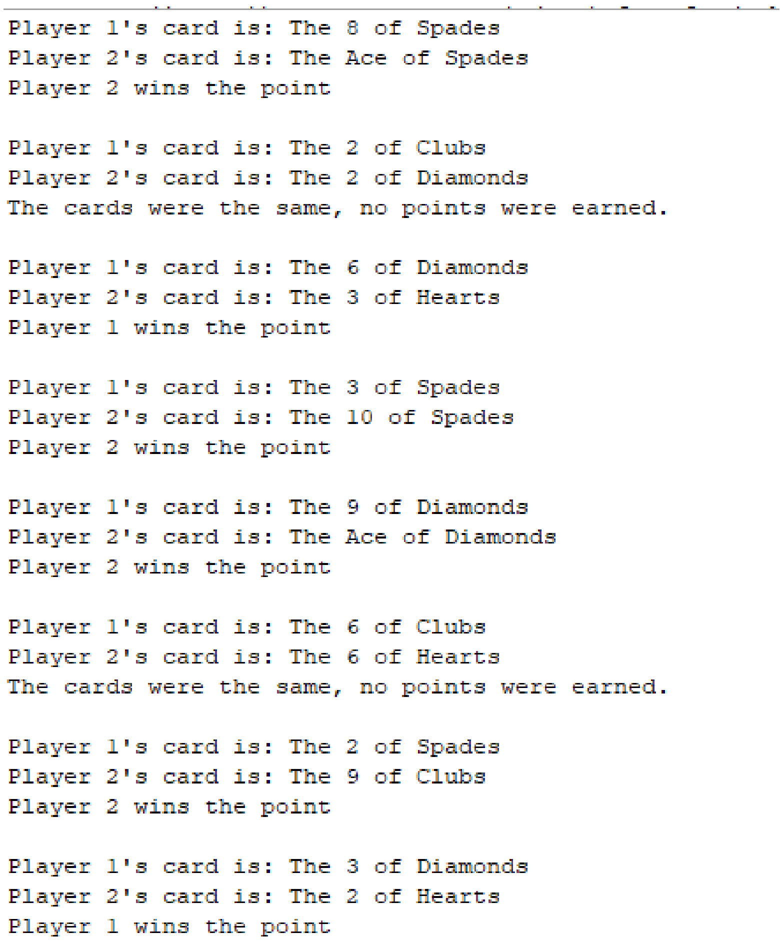


```java
        //have players flip through the deck and see who wins each turn.
        for(int i = 0; i < 26; i++) {
            Card player1Card = player1.flip();
            Card player2Card = player2.flip();
            if( player1Card.getCardValue() > player2Card.getCardValue()) {
                player1.incrementScore();
                System.out.println("Player 1's card is: " + player1Card.describeCard());
                System.out.println("Player 2's card is: " + player2Card.describeCard());
                System.out.println("Player 1 wins the point");
                System.out.println();}
            else if( player1Card.getCardValue() < player2Card.getCardValue()) {
                player2.incrementScore();
                System.out.println("Player 1's card is: " + player1Card.describeCard());
                System.out.println("Player 2's card is: " + player2Card.describeCard());
                System.out.println("Player 2 wins the point");
                System.out.println();}
            else if ( player1Card.getCardValue() == player2Card.getCardValue()) {
                System.out.println("Player 1's card is: " + player1Card.describeCard());
                System.out.println("Player 2's card is: " + player2Card.describeCard());
                System.out.println("The cards were the same, no points were earned.");
                System.out.println();

            }else {
                System.out.println("***");
            }
        }
            //Print out final scores!
            System.out.println(player1.describe());
            System.out.println(player2.describe());
```

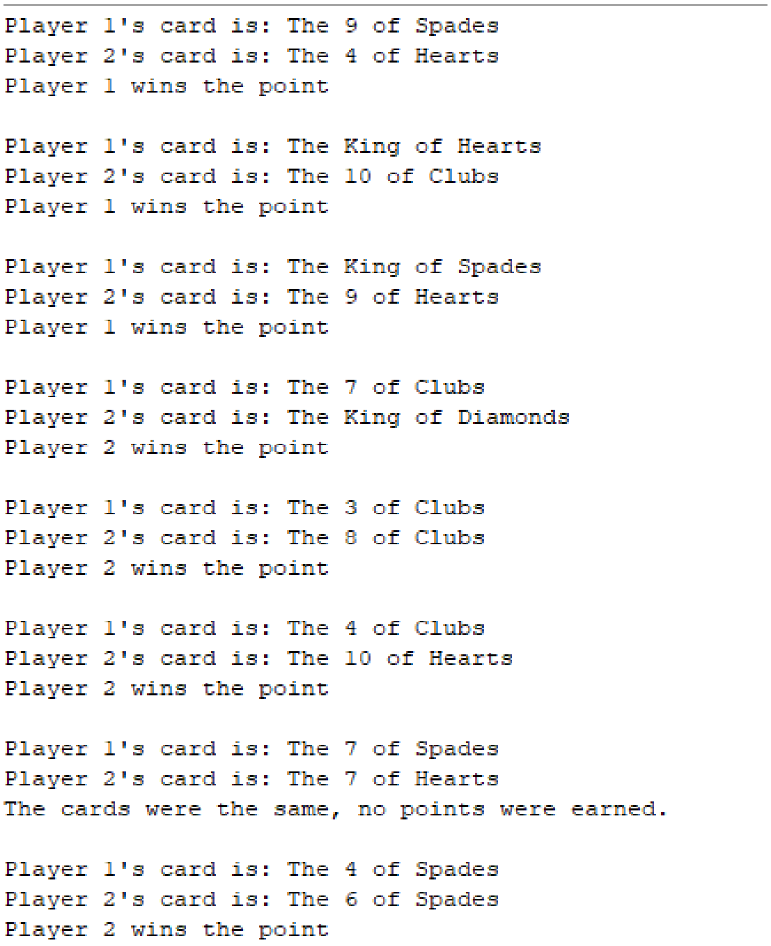


```java

            if(player1.playerScore() > player2.playerScore()) {
                System.out.println("Player 1 Has won the game!!");
            }if(player1.playerScore() < player2.playerScore()) {
                System.out.println("Player 2 Has won the game!!");
            }else if (player1.playerScore() == player2.playerScore()) {
                System.out.println("The game has ended in a draw");
            }
        }

    }


```

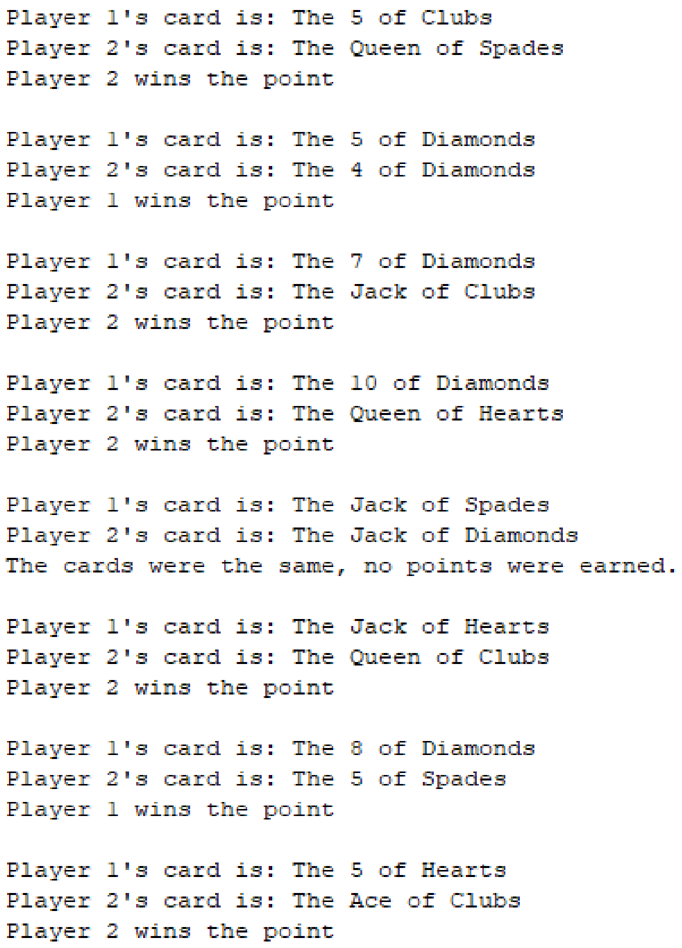
Screenshots of Running Application:



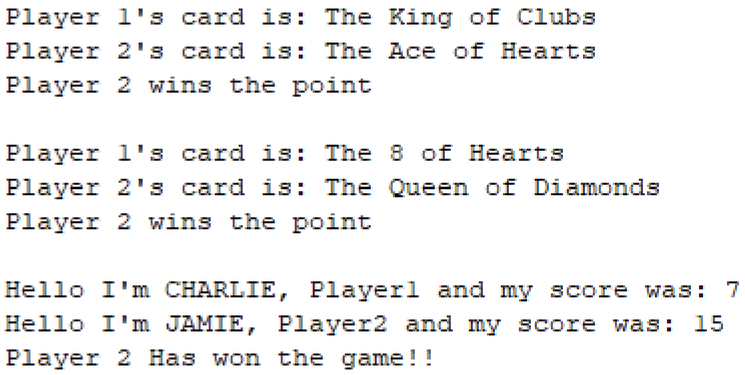
```
Player 1's card is: The 8 of Spades
Player 2's card is: The Ace of Spades
Player 2 wins the point

Player 1's card is: The 2 of Clubs
Player 2's card is: The 2 of Diamonds
The cards were the same, no points were earned.

Player 1's card is: The 6 of Diamonds
Player 2's card is: The 3 of Hearts
Player 1 wins the point

Player 1's card is: The 3 of Spades
Player 2's card is: The 10 of Spades
Player 2 wins the point

Player 1's card is: The 9 of Diamonds
Player 2's card is: The Ace of Diamonds
Player 2 wins the point

Player 1's card is: The 6 of Clubs
Player 2's card is: The 6 of Hearts
The cards were the same, no points were earned.

Player 1's card is: The 2 of Spades
Player 2's card is: The 9 of Clubs
Player 2 wins the point

Player 1's card is: The 3 of Diamonds
Player 2's card is: The 2 of Hearts
Player 1 wins the point
```



```
Player 1's card is: The 9 of Spades
Player 2's card is: The 4 of Hearts
Player 1 wins the point

Player 1's card is: The King of Hearts
Player 2's card is: The 10 of Clubs
Player 1 wins the point

Player 1's card is: The King of Spades
Player 2's card is: The 9 of Hearts
Player 1 wins the point

Player 1's card is: The 7 of Clubs
Player 2's card is: The King of Diamonds
Player 2 wins the point

Player 1's card is: The 3 of Clubs
Player 2's card is: The 8 of Clubs
Player 2 wins the point

Player 1's card is: The 4 of Clubs
Player 2's card is: The 10 of Hearts
Player 2 wins the point

Player 1's card is: The 7 of Spades
Player 2's card is: The 7 of Hearts
The cards were the same, no points were earned.

Player 1's card is: The 4 of Spades
Player 2's card is: The 6 of Spades
Player 2 wins the point
```



```
Player 1's card is: The 5 of Clubs
Player 2's card is: The Queen of Spades
Player 2 wins the point

Player 1's card is: The 5 of Diamonds
Player 2's card is: The 4 of Diamonds
Player 1 wins the point

Player 1's card is: The 7 of Diamonds
Player 2's card is: The Jack of Clubs
Player 2 wins the point

Player 1's card is: The 10 of Diamonds
Player 2's card is: The Queen of Hearts
Player 2 wins the point

Player 1's card is: The Jack of Spades
Player 2's card is: The Jack of Diamonds
The cards were the same, no points were earned.

Player 1's card is: The Jack of Hearts
Player 2's card is: The Queen of Clubs
Player 2 wins the point

Player 1's card is: The 8 of Diamonds
Player 2's card is: The 5 of Spades
Player 1 wins the point

Player 1's card is: The 5 of Hearts
Player 2's card is: The Ace of Clubs
Player 2 wins the point
```



```
Player 1's card is: The King of Clubs
Player 2's card is: The Ace of Hearts
Player 2 wins the point

Player 1's card is: The 8 of Hearts
Player 2's card is: The Queen of Diamonds
Player 2 wins the point

Hello I'm CHARLIE, Player1 and my score was: 7
Hello I'm JAMIE, Player2 and my score was: 15
Player 2 Has won the game!!
```

URL to GitHub Repository:

https://github.com/DGolf1313/Week6CodingAssignment