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.

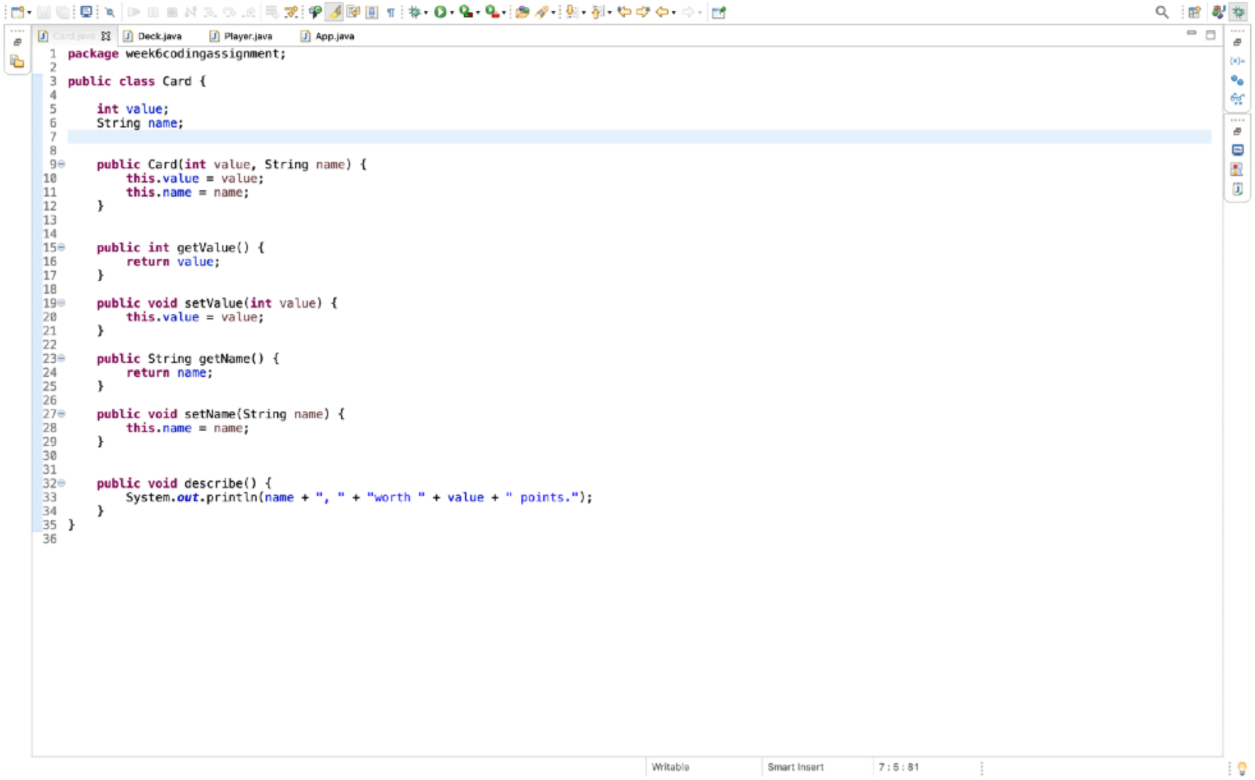
c. Player
   i. Fields
      1. **hand** (List of Card)
      2. score (set to 0 in the constructor)
      3. name
   ii. Methods
      1. **describe** (prints out information about the player and calls the describe method for each card in the Hand List)
      2. **flip** (removes and returns the top 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.

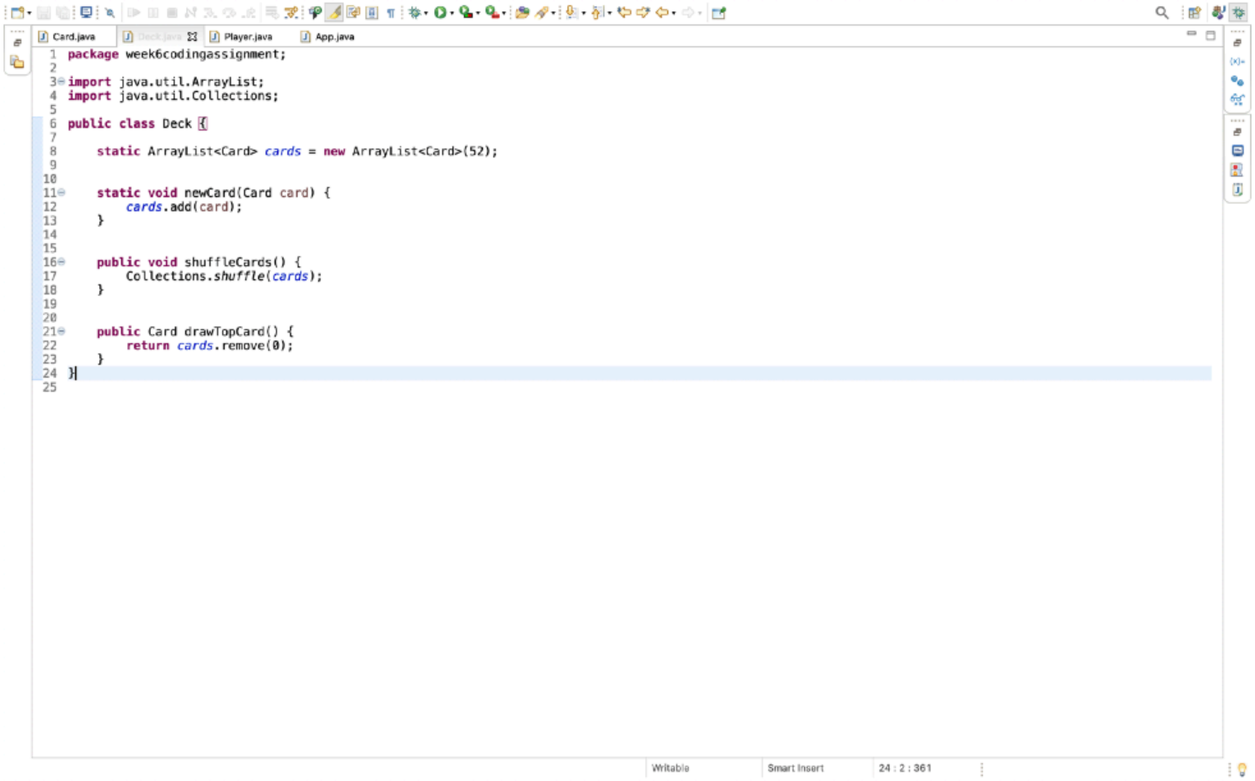
    a. Compare the value of each card returned by the two player’s flip methods. Call the incrementScore method on the player whose card has the higher value.

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.

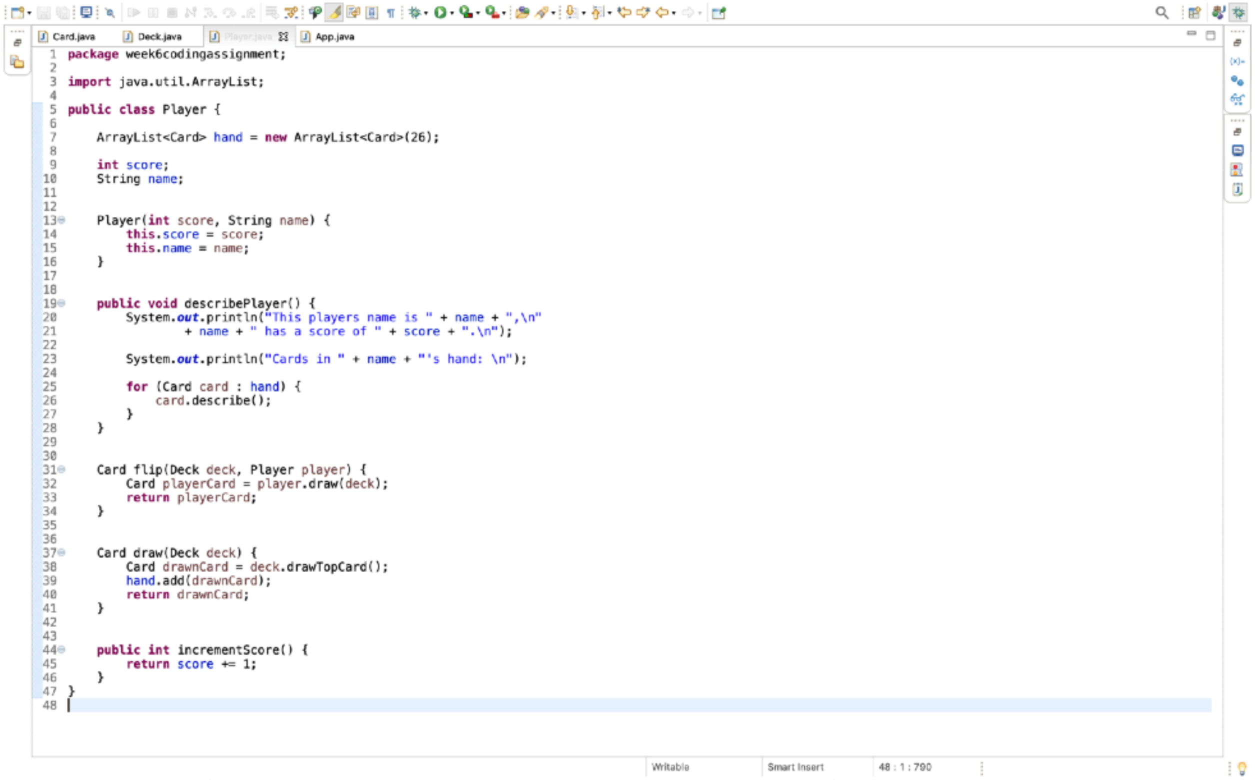
Screenshots of Code:



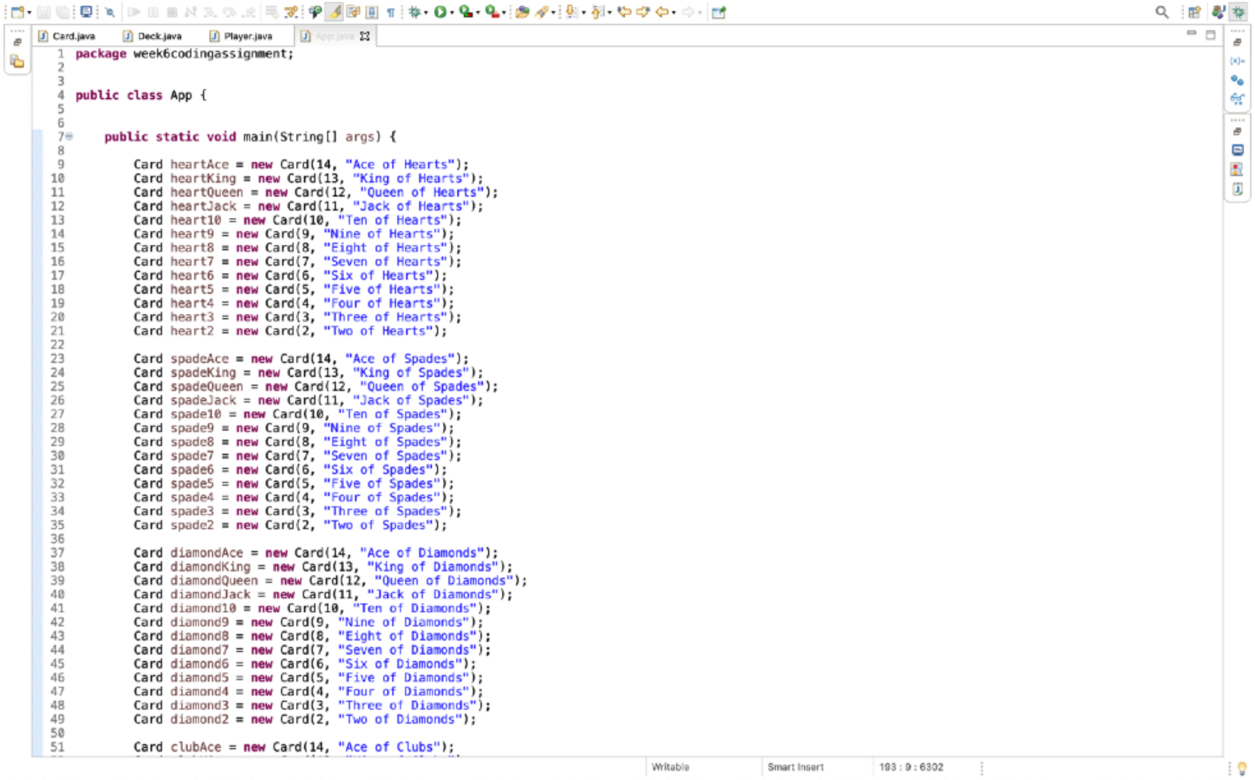
```java
package week6codingassignment;

public class Card {

    int value;
    String name;


    public Card(int value, String name) {
        this.value = value;
        this.name = name;
    }


    public int getValue() {
        return value;
    }

    public void setValue(int value) {
        this.value = value;
    }

    public String getName() {
        return name;
    }

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


    public void describe() {
        System.out.println(name + ", " + "worth " + value + " points.");
    }
}

```



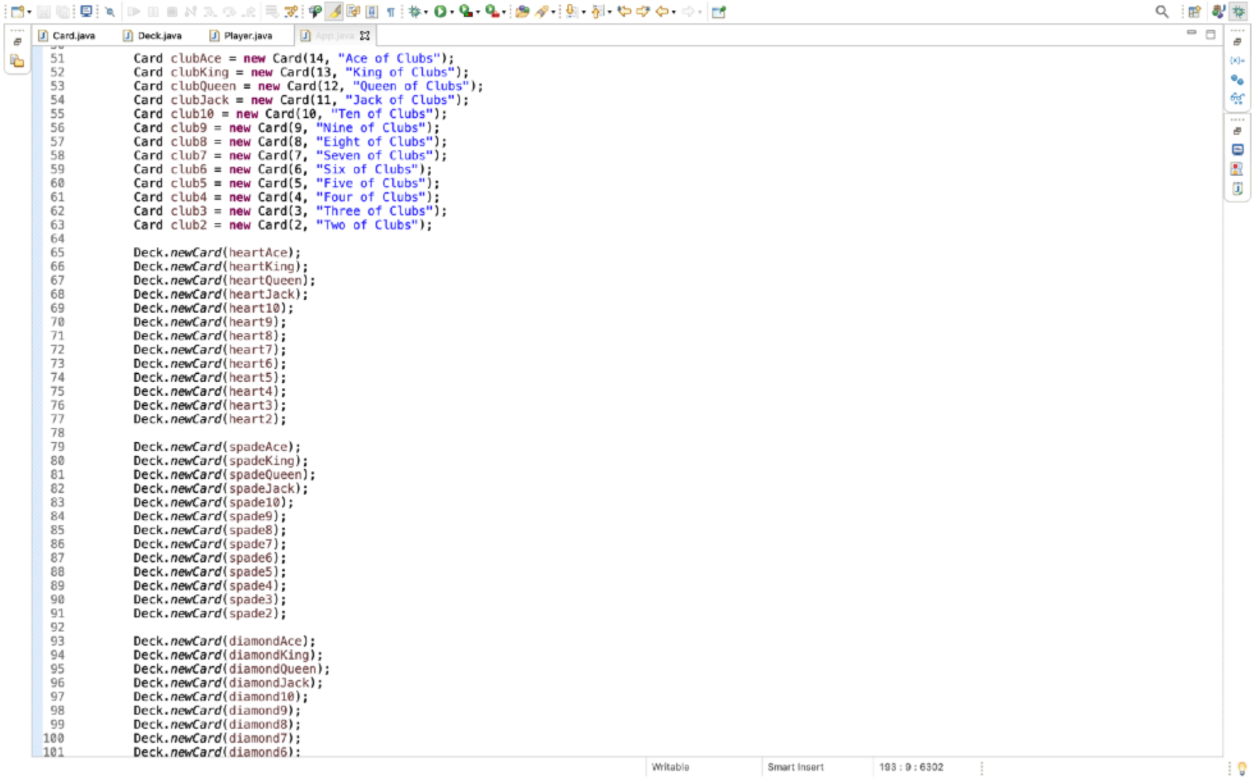
```java
package week6codingassignment;

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

public class Deck {

    static ArrayList<Card> cards = new ArrayList<Card>(52);


    static void newCard(Card card) {
        cards.add(card);
    }


    public void shuffleCards() {
        Collections.shuffle(cards);
    }


    public Card drawTopCard() {
        return cards.remove(0);
    }
}

```



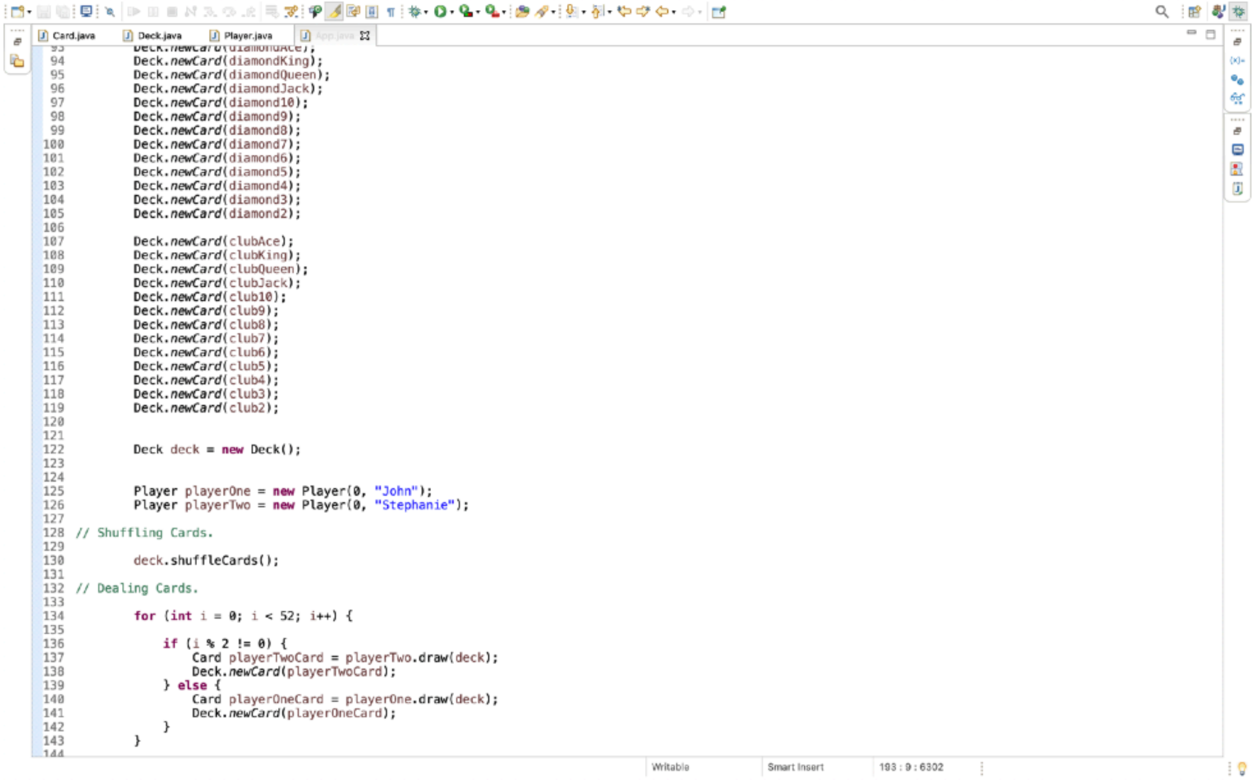
```java
package week6codingassignment;

import java.util.ArrayList;

public class Player {

    ArrayList<Card> hand = new ArrayList<Card>(26);

    int score;
    String name;


    Player(int score, String name) {
        this.score = score;
        this.name = name;
    }


    public void describePlayer() {
        System.out.println("This players name is " + name + ",\n"
                + name + " has a score of " + score + ".\n");

        System.out.println("Cards in " + name + "'s hand: \n");

        for (Card card : hand) {
            card.describe();
        }
    }


    Card flip(Deck deck, Player player) {
        Card playerCard = player.draw(deck);
        return playerCard;
    }


    Card draw(Deck deck) {
        Card drawnCard = deck.drawTopCard();
        hand.add(drawnCard);
        return drawnCard;
    }


    public int incrementScore() {
        return score += 1;
    }
}

```



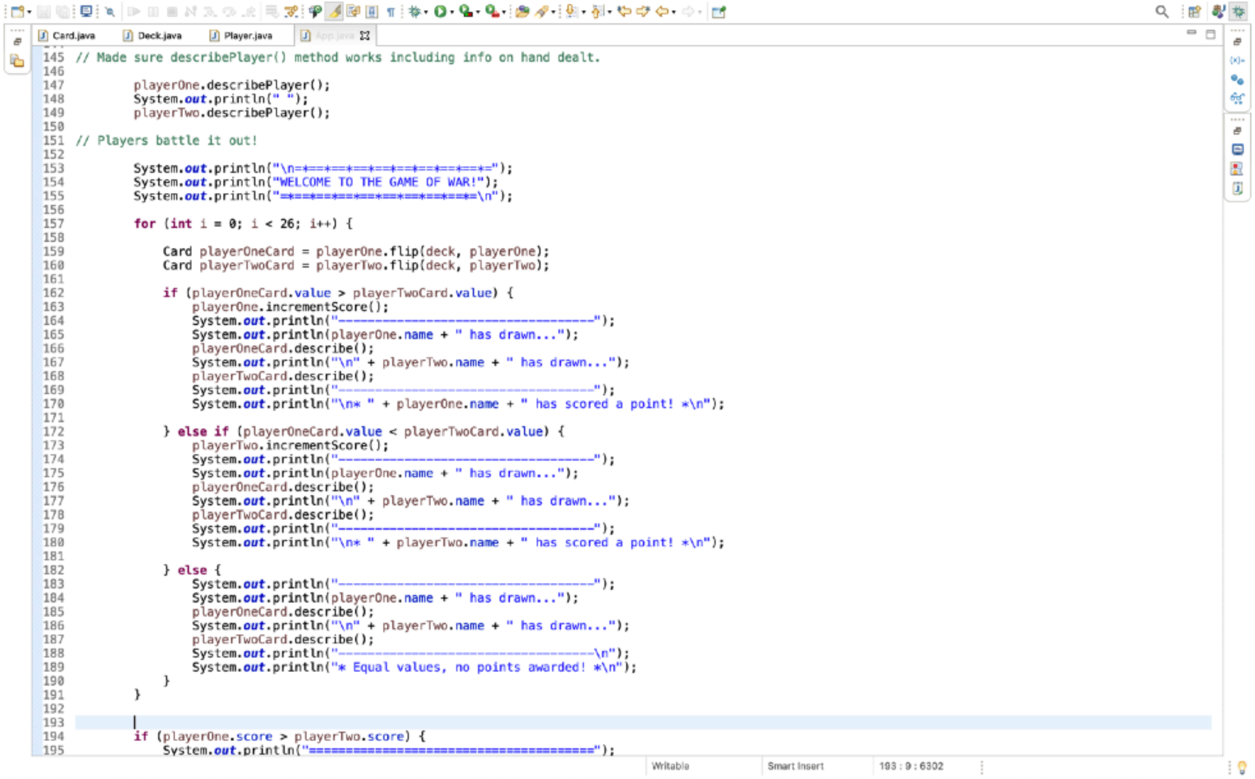
```
package week6codingassignment;


public class App {


    public static void main(String[] args) {

        Card heartAce = new Card(14, "Ace of Hearts");
        Card heartKing = new Card(13, "King of Hearts");
        Card heartQueen = new Card(12, "Queen of Hearts");
        Card heartJack = new Card(11, "Jack of Hearts");
        Card heart10 = new Card(10, "Ten of Hearts");
        Card heart9 = new Card(9, "Nine of Hearts");
        Card heart8 = new Card(8, "Eight of Hearts");
        Card heart7 = new Card(7, "Seven of Hearts");
        Card heart6 = new Card(6, "Six of Hearts");
        Card heart5 = new Card(5, "Five of Hearts");
        Card heart4 = new Card(4, "Four of Hearts");
        Card heart3 = new Card(3, "Three of Hearts");
        Card heart2 = new Card(2, "Two of Hearts");

        Card spadeAce = new Card(14, "Ace of Spades");
        Card spadeKing = new Card(13, "King of Spades");
        Card spadeQueen = new Card(12, "Queen of Spades");
        Card spadeJack = new Card(11, "Jack of Spades");
        Card spade10 = new Card(10, "Ten of Spades");
        Card spade9 = new Card(9, "Nine of Spades");
        Card spade8 = new Card(8, "Eight of Spades");
        Card spade7 = new Card(7, "Seven of Spades");
        Card spade6 = new Card(6, "Six of Spades");
        Card spade5 = new Card(5, "Five of Spades");
        Card spade4 = new Card(4, "Four of Spades");
        Card spade3 = new Card(3, "Three of Spades");
        Card spade2 = new Card(2, "Two of Spades");

        Card diamondAce = new Card(14, "Ace of Diamonds");
        Card diamondKing = new Card(13, "King of Diamonds");
        Card diamondQueen = new Card(12, "Queen of Diamonds");
        Card diamondJack = new Card(11, "Jack of Diamonds");
        Card diamond10 = new Card(10, "Ten of Diamonds");
        Card diamond9 = new Card(9, "Nine of Diamonds");
        Card diamond8 = new Card(8, "Eight of Diamonds");
        Card diamond7 = new Card(7, "Seven of Diamonds");
        Card diamond6 = new Card(6, "Six of Diamonds");
        Card diamond5 = new Card(5, "Five of Diamonds");
        Card diamond4 = new Card(4, "Four of Diamonds");
        Card diamond3 = new Card(3, "Three of Diamonds");
        Card diamond2 = new Card(2, "Two of Diamonds");

        Card clubAce = new Card(14, "Ace of Clubs");
```



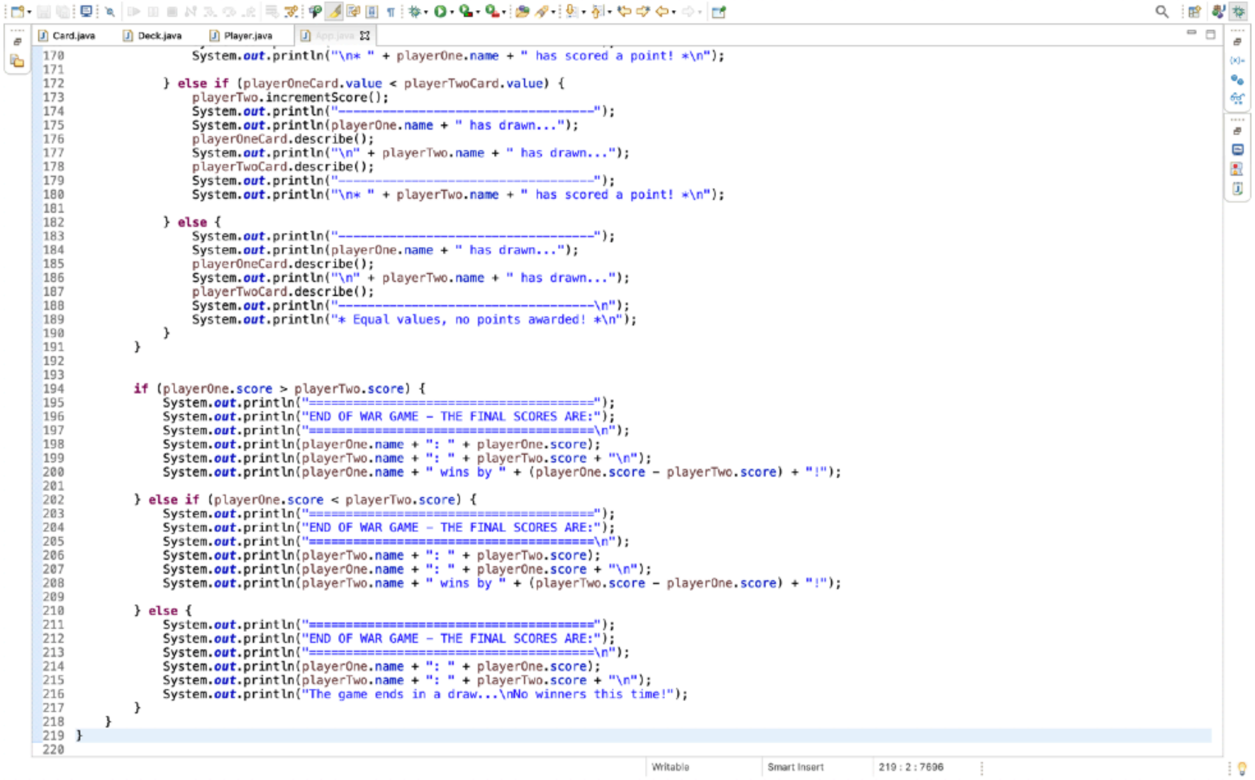
```
        Card clubAce = new Card(14, "Ace of Clubs");
        Card clubKing = new Card(13, "King of Clubs");
        Card clubQueen = new Card(12, "Queen of Clubs");
        Card clubJack = new Card(11, "Jack of Clubs");
        Card club10 = new Card(10, "Ten of Clubs");
        Card club9 = new Card(9, "Nine of Clubs");
        Card club8 = new Card(8, "Eight of Clubs");
        Card club7 = new Card(7, "Seven of Clubs");
        Card club6 = new Card(6, "Six of Clubs");
        Card club5 = new Card(5, "Five of Clubs");
        Card club4 = new Card(4, "Four of Clubs");
        Card club3 = new Card(3, "Three of Clubs");
        Card club2 = new Card(2, "Two of Clubs");

        Deck.newCard(heartAce);
        Deck.newCard(heartKing);
        Deck.newCard(heartQueen);
        Deck.newCard(heartJack);
        Deck.newCard(heart10);
        Deck.newCard(heart9);
        Deck.newCard(heart8);
        Deck.newCard(heart7);
        Deck.newCard(heart6);
        Deck.newCard(heart5);
        Deck.newCard(heart4);
        Deck.newCard(heart3);
        Deck.newCard(heart2);

        Deck.newCard(spadeAce);
        Deck.newCard(spadeKing);
        Deck.newCard(spadeQueen);
        Deck.newCard(spadeJack);
        Deck.newCard(spade10);
        Deck.newCard(spade9);
        Deck.newCard(spade8);
        Deck.newCard(spade7);
        Deck.newCard(spade6);
        Deck.newCard(spade5);
        Deck.newCard(spade4);
        Deck.newCard(spade3);
        Deck.newCard(spade2);

        Deck.newCard(diamondAce);
        Deck.newCard(diamondKing);
        Deck.newCard(diamondQueen);
        Deck.newCard(diamondJack);
        Deck.newCard(diamond10);
        Deck.newCard(diamond9);
        Deck.newCard(diamond8);
        Deck.newCard(diamond7);
        Deck.newCard(diamond6);
```



```java
        Deck.newCard(diamondAce);
        Deck.newCard(diamondKing);
        Deck.newCard(diamondQueen);
        Deck.newCard(diamondJack);
        Deck.newCard(diamond10);
        Deck.newCard(diamond9);
        Deck.newCard(diamond8);
        Deck.newCard(diamond7);
        Deck.newCard(diamond6);
        Deck.newCard(diamond5);
        Deck.newCard(diamond4);
        Deck.newCard(diamond3);
        Deck.newCard(diamond2);

        Deck.newCard(clubAce);
        Deck.newCard(clubKing);
        Deck.newCard(clubQueen);
        Deck.newCard(clubJack);
        Deck.newCard(club10);
        Deck.newCard(club9);
        Deck.newCard(club8);
        Deck.newCard(club7);
        Deck.newCard(club6);
        Deck.newCard(club5);
        Deck.newCard(club4);
        Deck.newCard(club3);
        Deck.newCard(club2);


        Deck deck = new Deck();


        Player playerOne = new Player(0, "John");
        Player playerTwo = new Player(0, "Stephanie");

// Shuffling Cards.

        deck.shuffleCards();

// Dealing Cards.

        for (int i = 0; i < 52; i++) {

            if (i % 2 != 0) {
                Card playerTwoCard = playerTwo.draw(deck);
                Deck.newCard(playerTwoCard);
            } else {
                Card playerOneCard = playerOne.draw(deck);
                Deck.newCard(playerOneCard);
            }
        }

```

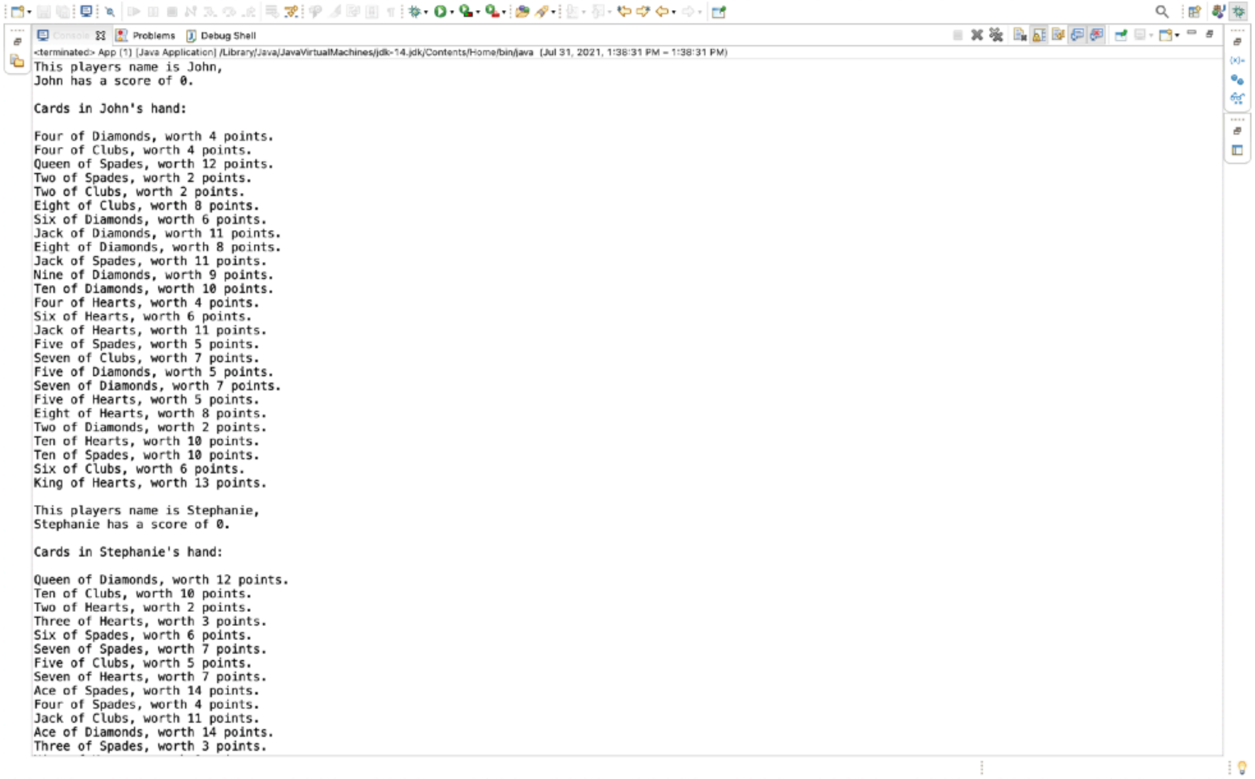


```java
// Made sure describePlayer() method works including info on hand dealt.

        playerOne.describePlayer();
        System.out.println(" ");
        playerTwo.describePlayer();

// Players battle it out!

        System.out.println("\n=*==*==*==*==*==*==*==*==*=");
        System.out.println("WELCOME TO THE GAME OF WAR!");
        System.out.println("=*==*==*==*==*==*==*==*==*=\n");

        for (int i = 0; i < 26; i++) {

            Card playerOneCard = playerOne.flip(deck, playerOne);
            Card playerTwoCard = playerTwo.flip(deck, playerTwo);

            if (playerOneCard.value > playerTwoCard.value) {
                playerOne.incrementScore();
                System.out.println("--------------------------------");
                System.out.println(playerOne.name + " has drawn...");
                playerOneCard.describe();
                System.out.println("\n" + playerTwo.name + " has drawn...");
                playerTwoCard.describe();
                System.out.println("--------------------------------");
                System.out.println("\n* " + playerOne.name + " has scored a point! *\n");

            } else if (playerOneCard.value < playerTwoCard.value) {
                playerTwo.incrementScore();
                System.out.println("--------------------------------");
                System.out.println(playerOne.name + " has drawn...");
                playerOneCard.describe();
                System.out.println("\n" + playerTwo.name + " has drawn...");
                playerTwoCard.describe();
                System.out.println("--------------------------------");
                System.out.println("\n* " + playerTwo.name + " has scored a point! *\n");

            } else {
                System.out.println("--------------------------------");
                System.out.println(playerOne.name + " has drawn...");
                playerOneCard.describe();
                System.out.println("\n" + playerTwo.name + " has drawn...");
                playerTwoCard.describe();
                System.out.println("--------------------------------\n");
                System.out.println("* Equal values, no points awarded! *\n");
            }
        }


        if (playerOne.score > playerTwo.score) {
            System.out.println("====================================");
```

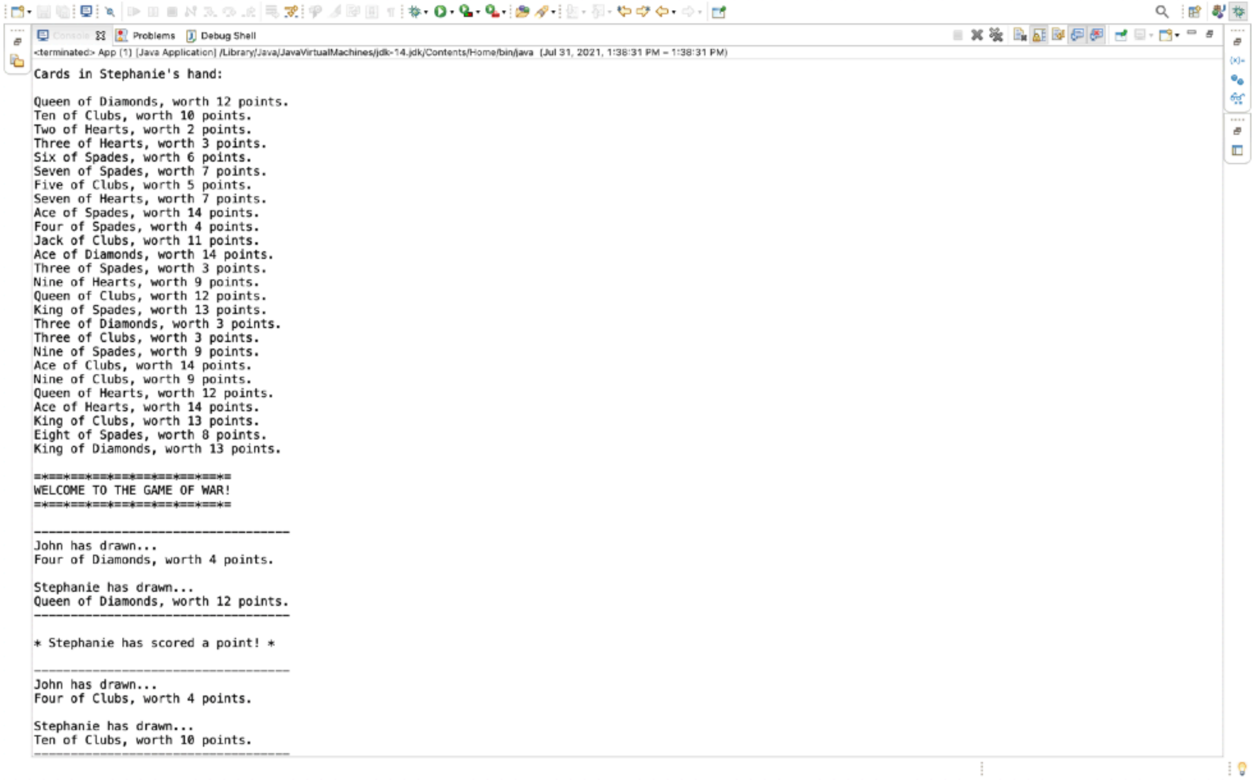


```java
                System.out.println("\n* " + playerOne.name + " has scored a point! *\n");

            } else if (playerOneCard.value < playerTwoCard.value) {
                playerTwo.incrementScore();
                System.out.println("------------------------------------");
                System.out.println(playerOne.name + " has drawn...");
                playerOneCard.describe();
                System.out.println("\n" + playerTwo.name + " has drawn...");
                playerTwoCard.describe();
                System.out.println("------------------------------------");
                System.out.println("\n* " + playerTwo.name + " has scored a point! *\n");

            } else {
                System.out.println("------------------------------------");
                System.out.println(playerOne.name + " has drawn...");
                playerOneCard.describe();
                System.out.println("\n" + playerTwo.name + " has drawn...");
                playerTwoCard.describe();
                System.out.println("------------------------------------\n");
                System.out.println("* Equal values, no points awarded! *\n");
            }
        }


        if (playerOne.score > playerTwo.score) {
            System.out.println("=======================================");
            System.out.println("END OF WAR GAME - THE FINAL SCORES ARE:");
            System.out.println("=======================================\n");
            System.out.println(playerOne.name + ": " + playerOne.score);
            System.out.println(playerTwo.name + ": " + playerTwo.score + "\n");
            System.out.println(playerOne.name + " wins by " + (playerOne.score - playerTwo.score) + "!");

        } else if (playerOne.score < playerTwo.score) {
            System.out.println("=======================================");
            System.out.println("END OF WAR GAME - THE FINAL SCORES ARE:");
            System.out.println("=======================================\n");
            System.out.println(playerTwo.name + ": " + playerTwo.score);
            System.out.println(playerOne.name + ": " + playerOne.score + "\n");
            System.out.println(playerTwo.name + " wins by " + (playerTwo.score - playerOne.score) + "!");

        } else {
            System.out.println("=======================================");
            System.out.println("END OF WAR GAME - THE FINAL SCORES ARE:");
            System.out.println("=======================================\n");
            System.out.println(playerOne.name + ": " + playerOne.score);
            System.out.println(playerTwo.name + ": " + playerTwo.score + "\n");
            System.out.println("The game ends in a draw...\nNo winners this time!");
        }
    }
}
```

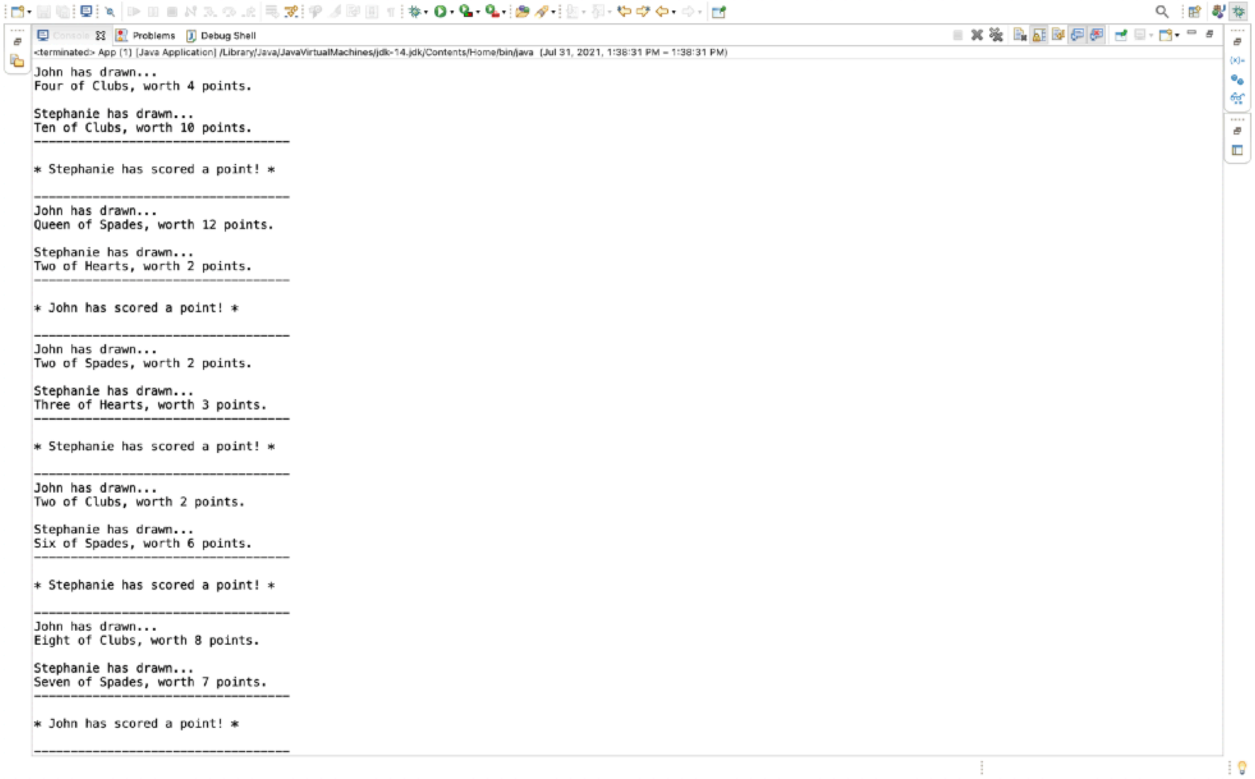
Screenshots of Running Application:



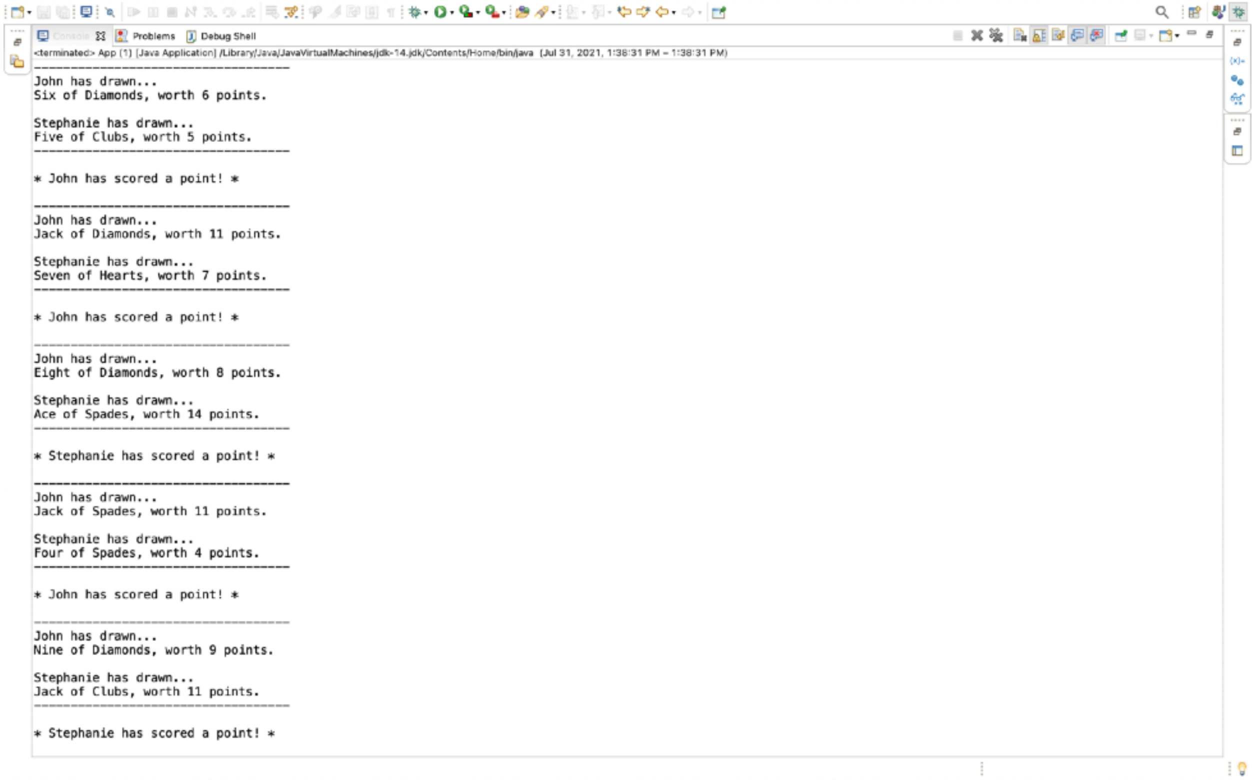
```
<terminated> App (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.jdk/Contents/Home/bin/java  (Jul 31, 2021, 1:38:31 PM – 1:38:31 PM)
This players name is John,
John has a score of 0.

Cards in John's hand:

Four of Diamonds, worth 4 points.
Four of Clubs, worth 4 points.
Queen of Spades, worth 12 points.
Two of Spades, worth 2 points.
Two of Clubs, worth 2 points.
Eight of Clubs, worth 8 points.
Six of Diamonds, worth 6 points.
Jack of Diamonds, worth 11 points.
Eight of Diamonds, worth 8 points.
Jack of Spades, worth 11 points.
Nine of Diamonds, worth 9 points.
Ten of Diamonds, worth 10 points.
Four of Hearts, worth 4 points.
Six of Hearts, worth 6 points.
Jack of Hearts, worth 11 points.
Five of Spades, worth 5 points.
Seven of Clubs, worth 7 points.
Five of Diamonds, worth 5 points.
Seven of Diamonds, worth 7 points.
Five of Hearts, worth 5 points.
Eight of Hearts, worth 8 points.
Two of Diamonds, worth 2 points.
Ten of Hearts, worth 10 points.
Ten of Spades, worth 10 points.
Six of Clubs, worth 6 points.
King of Hearts, worth 13 points.

This players name is Stephanie,
Stephanie has a score of 0.

Cards in Stephanie's hand:

Queen of Diamonds, worth 12 points.
Ten of Clubs, worth 10 points.
Two of Hearts, worth 2 points.
Three of Hearts, worth 3 points.
Six of Spades, worth 6 points.
Seven of Spades, worth 7 points.
Five of Clubs, worth 5 points.
Seven of Hearts, worth 7 points.
Ace of Spades, worth 14 points.
Four of Spades, worth 4 points.
Jack of Clubs, worth 11 points.
Ace of Diamonds, worth 14 points.
Three of Spades, worth 3 points.
```



```
<terminated> App (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.jdk/Contents/Home/bin/java (Jul 31, 2021, 1:38:31 PM – 1:38:31 PM)
Cards in Stephanie's hand:

Queen of Diamonds, worth 12 points.
Ten of Clubs, worth 10 points.
Two of Hearts, worth 2 points.
Three of Hearts, worth 3 points.
Six of Spades, worth 6 points.
Seven of Spades, worth 7 points.
Five of Clubs, worth 5 points.
Seven of Hearts, worth 7 points.
Ace of Spades, worth 14 points.
Four of Spades, worth 4 points.
Jack of Clubs, worth 11 points.
Ace of Diamonds, worth 14 points.
Three of Spades, worth 3 points.
Nine of Hearts, worth 9 points.
Queen of Clubs, worth 12 points.
King of Spades, worth 13 points.
Three of Diamonds, worth 3 points.
Three of Clubs, worth 3 points.
Nine of Spades, worth 9 points.
Ace of Clubs, worth 14 points.
Nine of Clubs, worth 9 points.
Queen of Hearts, worth 12 points.
Ace of Hearts, worth 14 points.
King of Clubs, worth 13 points.
Eight of Spades, worth 8 points.
King of Diamonds, worth 13 points.

=*==*==*==*==*==*==*==*==*=
WELCOME TO THE GAME OF WAR!
=*==*==*==*==*==*==*==*==*=

-----------------------------------
John has drawn...
Four of Diamonds, worth 4 points.

Stephanie has drawn...
Queen of Diamonds, worth 12 points.
-----------------------------------

* Stephanie has scored a point! *

-----------------------------------
John has drawn...
Four of Clubs, worth 4 points.

Stephanie has drawn...
Ten of Clubs, worth 10 points.
-----------------------------------
```

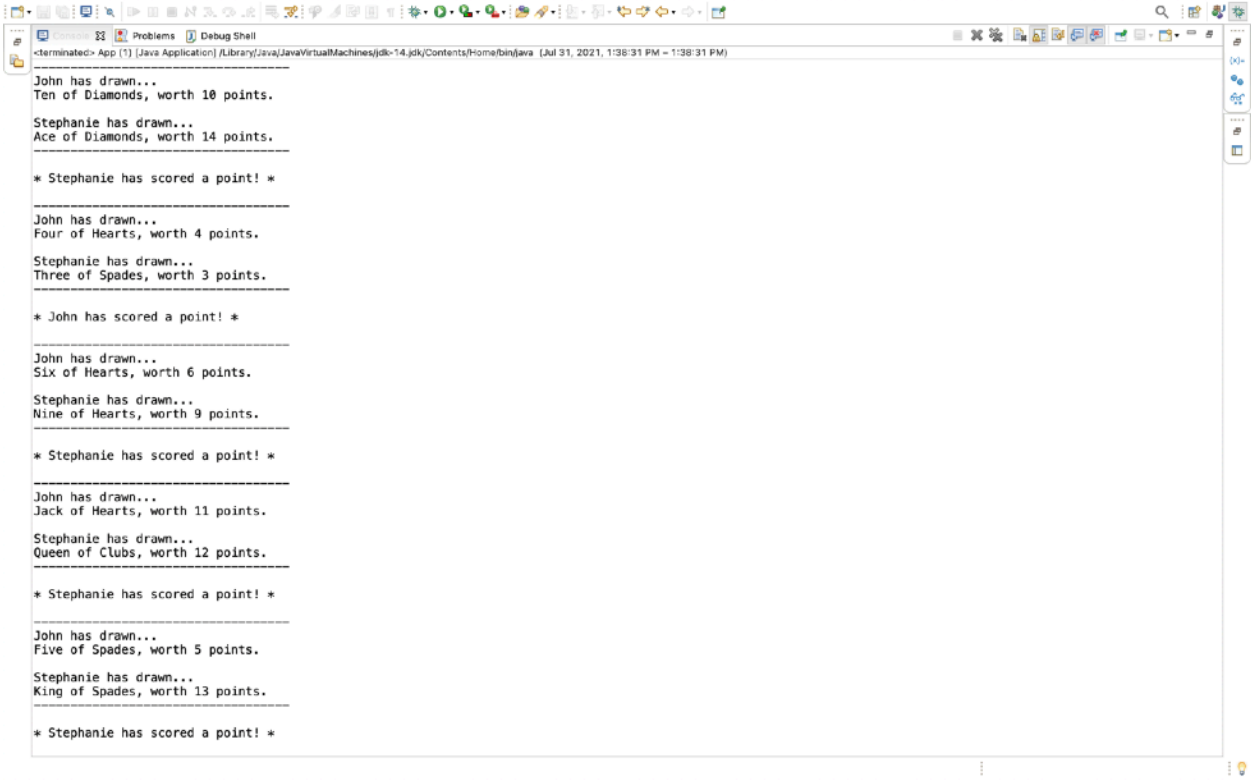


```
<terminated> App (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.jdk/Contents/Home/bin/java (Jul 31, 2021, 1:38:31 PM – 1:38:31 PM)
John has drawn...
Four of Clubs, worth 4 points.

Stephanie has drawn...
Ten of Clubs, worth 10 points.
-----------------------------------

* Stephanie has scored a point! *

-----------------------------------
John has drawn...
Queen of Spades, worth 12 points.

Stephanie has drawn...
Two of Hearts, worth 2 points.
-----------------------------------

* John has scored a point! *

-----------------------------------
John has drawn...
Two of Spades, worth 2 points.

Stephanie has drawn...
Three of Hearts, worth 3 points.
-----------------------------------

* Stephanie has scored a point! *

-----------------------------------
John has drawn...
Two of Clubs, worth 2 points.

Stephanie has drawn...
Six of Spades, worth 6 points.
-----------------------------------

* Stephanie has scored a point! *

-----------------------------------
John has drawn...
Eight of Clubs, worth 8 points.

Stephanie has drawn...
Seven of Spades, worth 7 points.
-----------------------------------

* John has scored a point! *

-----------------------------------
```



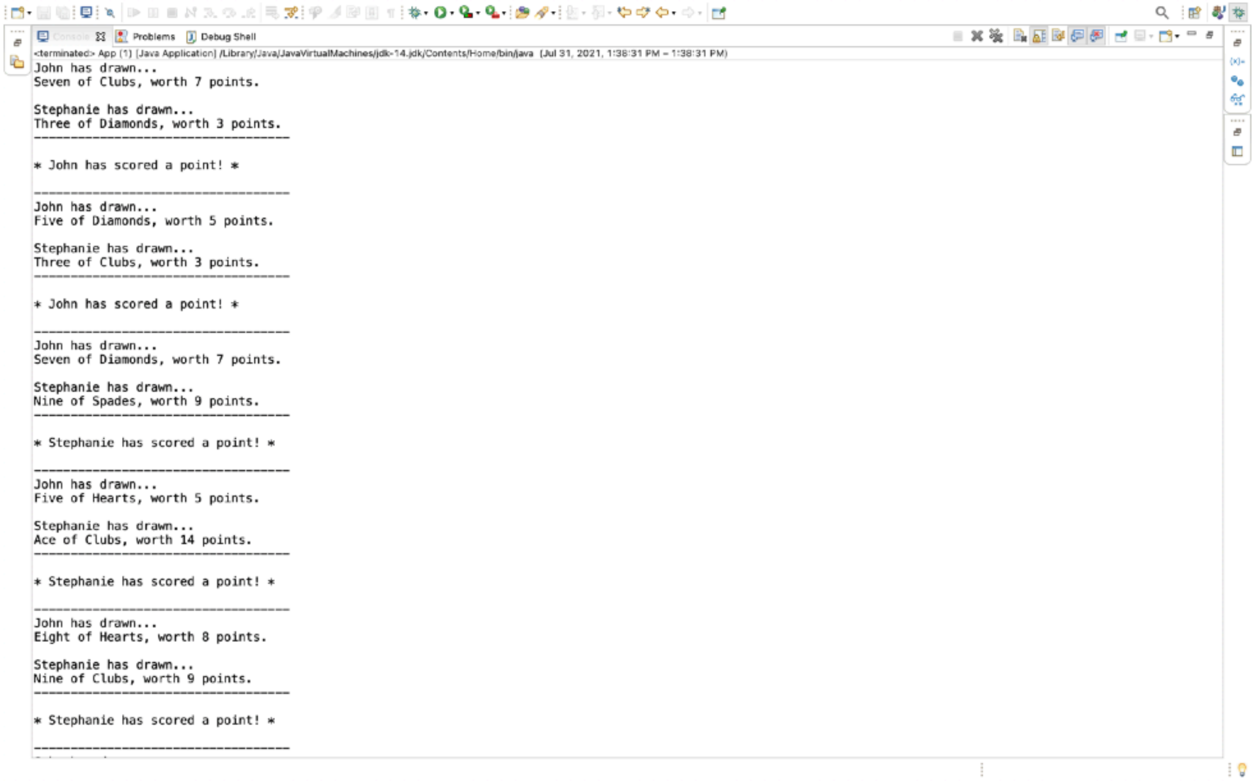
<terminated> App (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.jdk/Contents/Home/bin/java (Jul 31, 2021, 1:38:31 PM – 1:38:31 PM)

```
-----------------------------------
John has drawn...
Six of Diamonds, worth 6 points.

Stephanie has drawn...
Five of Clubs, worth 5 points.
-----------------------------------

* John has scored a point! *

-----------------------------------
John has drawn...
Jack of Diamonds, worth 11 points.

Stephanie has drawn...
Seven of Hearts, worth 7 points.
-----------------------------------

* John has scored a point! *

-----------------------------------
John has drawn...
Eight of Diamonds, worth 8 points.

Stephanie has drawn...
Ace of Spades, worth 14 points.
-----------------------------------

* Stephanie has scored a point! *

-----------------------------------
John has drawn...
Jack of Spades, worth 11 points.

Stephanie has drawn...
Four of Spades, worth 4 points.
-----------------------------------

* John has scored a point! *

-----------------------------------
John has drawn...
Nine of Diamonds, worth 9 points.

Stephanie has drawn...
Jack of Clubs, worth 11 points.
-----------------------------------

* Stephanie has scored a point! *
```

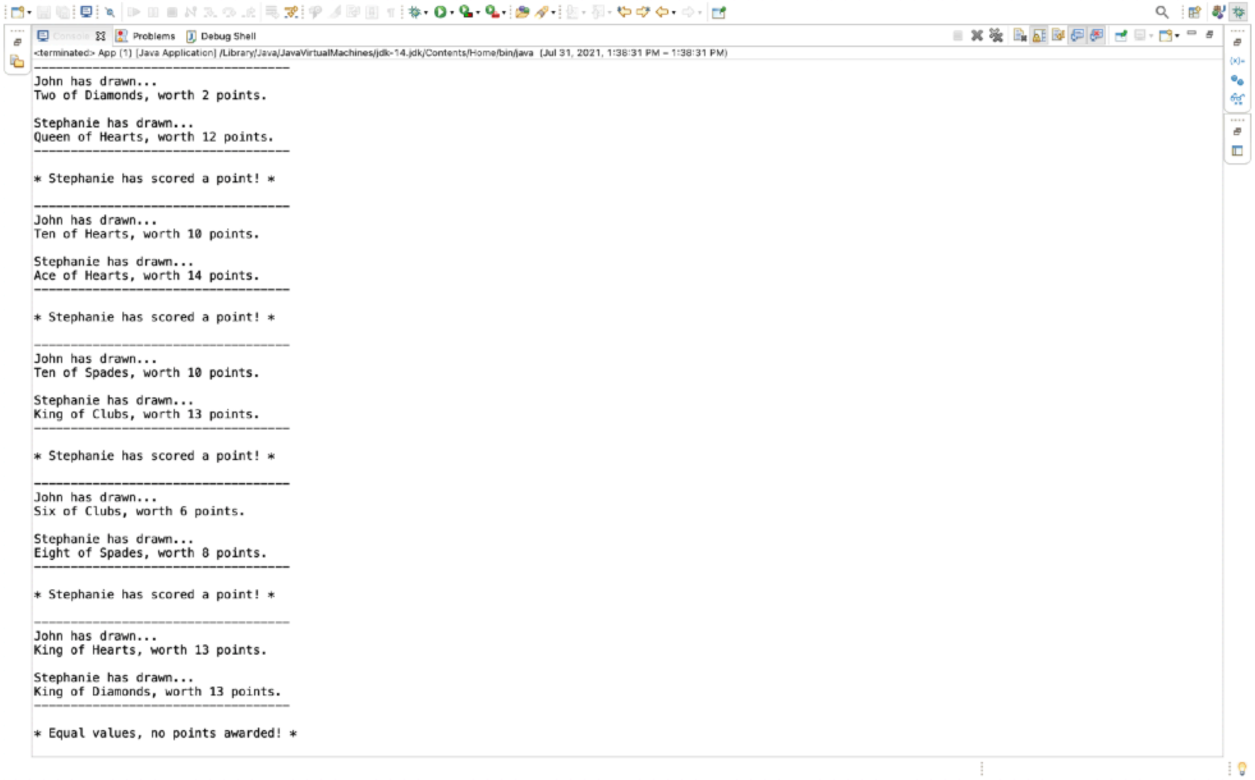


<terminated> App (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.jdk/Contents/Home/bin/java (Jul 31, 2021, 1:38:31 PM – 1:38:31 PM)

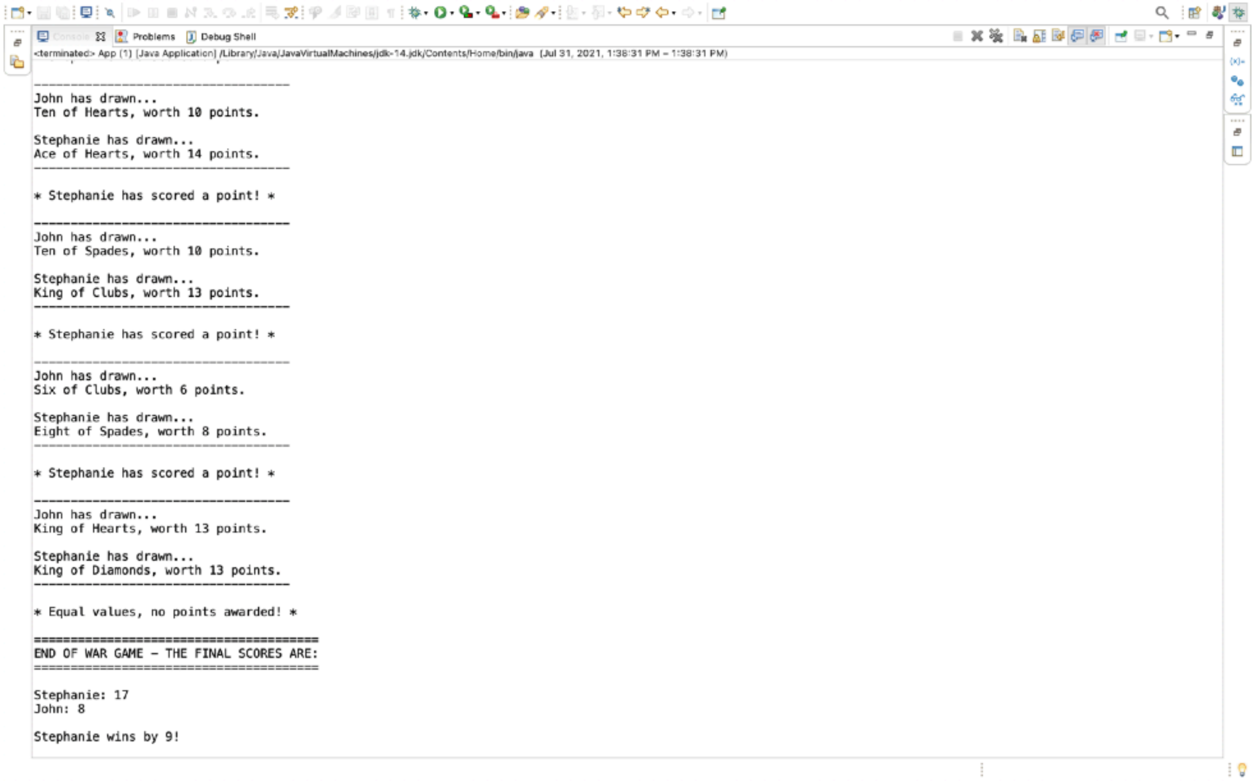
```
-----------------------------------
John has drawn...
Ten of Diamonds, worth 10 points.

Stephanie has drawn...
Ace of Diamonds, worth 14 points.
-----------------------------------

* Stephanie has scored a point! *

-----------------------------------
John has drawn...
Four of Hearts, worth 4 points.

Stephanie has drawn...
Three of Spades, worth 3 points.
-----------------------------------

* John has scored a point! *

-----------------------------------
John has drawn...
Six of Hearts, worth 6 points.

Stephanie has drawn...
Nine of Hearts, worth 9 points.
-----------------------------------

* Stephanie has scored a point! *

-----------------------------------
John has drawn...
Jack of Hearts, worth 11 points.

Stephanie has drawn...
Queen of Clubs, worth 12 points.
-----------------------------------

* Stephanie has scored a point! *

-----------------------------------
John has drawn...
Five of Spades, worth 5 points.

Stephanie has drawn...
King of Spades, worth 13 points.
-----------------------------------

* Stephanie has scored a point! *
```



```
<terminated> App (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.jdk/Contents/Home/bin/java (Jul 31, 2021, 1:38:31 PM – 1:38:31 PM)
John has drawn...
Seven of Clubs, worth 7 points.

Stephanie has drawn...
Three of Diamonds, worth 3 points.
-----------------------------------

* John has scored a point! *

-----------------------------------
John has drawn...
Five of Diamonds, worth 5 points.

Stephanie has drawn...
Three of Clubs, worth 3 points.
-----------------------------------

* John has scored a point! *

-----------------------------------
John has drawn...
Seven of Diamonds, worth 7 points.

Stephanie has drawn...
Nine of Spades, worth 9 points.
-----------------------------------

* Stephanie has scored a point! *

-----------------------------------
John has drawn...
Five of Hearts, worth 5 points.

Stephanie has drawn...
Ace of Clubs, worth 14 points.
-----------------------------------

* Stephanie has scored a point! *

-----------------------------------
John has drawn...
Eight of Hearts, worth 8 points.

Stephanie has drawn...
Nine of Clubs, worth 9 points.
-----------------------------------

* Stephanie has scored a point! *

-----------------------------------
```



```
<terminated> App (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.jdk/Contents/Home/bin/java (Jul 31, 2021, 1:38:31 PM – 1:38:31 PM)
-----------------------------------
John has drawn...
Two of Diamonds, worth 2 points.

Stephanie has drawn...
Queen of Hearts, worth 12 points.
-----------------------------------

* Stephanie has scored a point! *

-----------------------------------
John has drawn...
Ten of Hearts, worth 10 points.

Stephanie has drawn...
Ace of Hearts, worth 14 points.
-----------------------------------

* Stephanie has scored a point! *

-----------------------------------
John has drawn...
Ten of Spades, worth 10 points.

Stephanie has drawn...
King of Clubs, worth 13 points.
-----------------------------------

* Stephanie has scored a point! *

-----------------------------------
John has drawn...
Six of Clubs, worth 6 points.

Stephanie has drawn...
Eight of Spades, worth 8 points.
-----------------------------------

* Stephanie has scored a point! *

-----------------------------------
John has drawn...
King of Hearts, worth 13 points.

Stephanie has drawn...
King of Diamonds, worth 13 points.
-----------------------------------

* Equal values, no points awarded! *
```



<terminated> App (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.jdk/Contents/Home/bin/java (Jul 31, 2021, 1:38:31 PM – 1:38:31 PM)

```
-----------------------------------
John has drawn...
Ten of Hearts, worth 10 points.

Stephanie has drawn...
Ace of Hearts, worth 14 points.
-----------------------------------

* Stephanie has scored a point! *

-----------------------------------
John has drawn...
Ten of Spades, worth 10 points.

Stephanie has drawn...
King of Clubs, worth 13 points.
-----------------------------------

* Stephanie has scored a point! *

-----------------------------------
John has drawn...
Six of Clubs, worth 6 points.

Stephanie has drawn...
Eight of Spades, worth 8 points.
-----------------------------------

* Stephanie has scored a point! *

-----------------------------------
John has drawn...
King of Hearts, worth 13 points.

Stephanie has drawn...
King of Diamonds, worth 13 points.
-----------------------------------

* Equal values, no points awarded! *

=======================================
END OF WAR GAME – THE FINAL SCORES ARE:
=======================================

Stephanie: 17
John: 8

Stephanie wins by 9!
```

URL to GitHub Repository:

https://github.com/johnbarts/week6codingassignment