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.java

gameOfWarWeek6/src/gameOfWarWeek6/Card.java

```
3  public class Card {
4    // created a integer that set the varible
5    public final int CLUBS = 0;
6    public final int DIAMONDS = 1;
7    public final int SPADES = 2;
8    public final int HEARTS = 3;
9
10   public final int JACK = 11;
11   public final int QUEEN = 12;
12   public final int KING = 13;
13   public final int ACE = 14;
14
15   private int cardName;
16   private int cardValue;
17
18   // set getters and settersS
19   public Card(int card, int value) {
20     this.setCardName(card);
21     this.setCardValue(value);
22
23   }
24
25   public int getCardName() {
26     return cardName;
27   }
28
29   public void setCardName(int card) {
30     this.cardName = card;
31   }
32
33   public int getCardValue() {
34     return cardValue;
35   }
36
37   public void setCardValue(int cardValue) {
38       this.cardValue = cardValue;
39     }
40
41
42   public void describe () {
43     System.out.println("card is" + this.toString());
44   }
45   // created a to string method to call on the describe.
46   public String toString() {
47     String suitAndRank = "";
48
49         if (cardValue == 2) {
50             suitAndRank += "Two";
51         } else if (cardValue == 3) {
52             suitAndRank += "Three";
53         } else if (cardValue == 4) {
54             suitAndRank += "Four";
55         } else if (cardValue == 5) {
56             suitAndRank += "Five";
57         } else if (cardValue == 6) {
58             suitAndRank += "Six";
59         } else if (cardValue == 7) {
```

*Card.java ×

```
37  public void setCardValue(int cardValue) {
38      this.cardValue = cardValue;
39    }
40
41
42  public void describe () {
43    System.out.println("card is" + this.toString());
44  }
45  // created a to string method to call on the describe.
46  public String toString() {
47    String suitAndRank = "";
48
49        if (cardValue == 2) {
50            suitAndRank += "Two";
51        } else if (cardValue == 3) {
52            suitAndRank += "Three";
53        } else if (cardValue == 4) {
54            suitAndRank += "Four";
55        } else if (cardValue == 5) {
56            suitAndRank += "Five";
57        } else if (cardValue == 6) {
58            suitAndRank += "Six";
59        } else if (cardValue == 7) {
60            suitAndRank += "Seven";
61        } else if (cardValue == 8) {
62            suitAndRank += "Eight";
63        } else if (cardValue == 9) {
64            suitAndRank += "Nine";
65        } else if (cardValue == 10) {
66            suitAndRank += "Ten";
67        } else if (cardValue == JACK) {
68            suitAndRank += "Jack";
69        } else if (cardValue == QUEEN ) {
70            suitAndRank += "Queen";
71        } else if (cardValue == KING) {
72            suitAndRank += "King";
73        } else if (cardValue == ACE) {
74            suitAndRank += "Ace";
75        } else {
76            suitAndRank += cardValue;
77        }
78
79        suitAndRank += " of ";
80
81        if (cardName == CLUBS) {
82          suitAndRank += "Clubs";
83        } else if (cardName == DIAMONDS) {
84          suitAndRank += "Diamonds";
85        } else if (cardName == HEARTS) {
86          suitAndRank+= "Hearts";
87        } else if (cardName == SPADES) {
88          suitAndRank += "Spades";
89        }
90    return suitAndRank;
91  }
92
93
```

Deck.java

```java
package gameOfWarWeek6;

import java.util.ArrayList;

public class Deck {

  List<Card> cardDeck = new ArrayList<Card>();
  // created a deck created a loop for cardName & cardValue.
  public Deck() {
    for(int i = 0; i< 4; i++) {
      for(int j = 2; j <= 14; j++) {
        cardDeck.add(new Card(i,j));
      }
    }

  }

  public List<Card> getCards(){
    return cardDeck;
  }

  public void setCards(List<Card>cardDeck) {
    this.cardDeck = cardDeck;

  }

  public void shuffle() {
    Collections.shuffle(cardDeck);
  }

  public Card draw() {
    Card card = cardDeck.get(0);
    cardDeck.remove(card);
    return card;
  }

}

```

Player.java

```java
package gameOfWarWeek6;

import java.util.ArrayList;

public class Player {
  List<Card> hand = new ArrayList<Card>();
  int playerScore;
  String playerName;


  public Player(String name) {
    this.playerScore = 0;
    this.playerName = name;
  }//end of Player method

  public void describe() {

    for(Card card : hand) {
      System.out.println(playerName +"'s card is a " + card);
      System.out.println("\n");

    }
  }// end of describe method

  public Card flip() {
    Card card= hand.get(0);
    hand.remove(card);
    return card ;
  }// end of Flip method

  public void draw(Deck deck) {
    Card card = deck.draw();
    hand.add(card);
  }// end of draw method

  public void incrementScore() {
    this.playerScore++;
  }
  public int getScore() {
    return playerScore;
  }

  public String getName() {
    return playerName;
  }
}
```

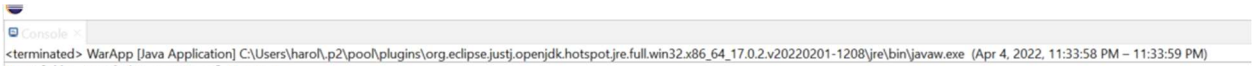
Player.java App.java

```java
package gameOfWarWeek6;

public class App {

  public static void main(String[] args) {

    // instantiated a deck, two players & called the shuffle method on the new deck.
    Deck newDeck = new Deck();

    newDeck.shuffle();

    Player player1 = new Player("Harold");
    Player player2 = new Player("Victoria");

    // Question 4 -created a for loop to iterate 52 items on the draw method
    for(int i = 0; i < 52; i++) {
      if(i % 2 == 0) {
        player1.hand.add(newDeck.draw());
      }else {
        player2.hand.add(newDeck.draw());
      }
    }

    player1.describe();
    player2.describe();

    // used a for loop to iterate 26 different times calling on the flip method.
    for(int i = 1; i<=26; i++) {
      Card player1Card = player1.flip();
      Card player2Card = player2.flip();
      System.out.println("ROUND" + ": " + i);

      // compared the players scores by calling on the getCardValue method.
      if(player1Card.getCardValue() > player2Card.getCardValue()) {
        player1.incrementScore();

      }else if(player2Card.getCardValue() > player1Card.getCardValue()) {
        player2.incrementScore();

      }else if(player1Card.getCardValue() == player2Card.getCardValue()) {
        player1.incrementScore();
        player2.incrementScore();

      }
      // comparing the scores between calling on the playerScore & playerName methods.
      if (player1.playerScore > player2.playerScore) {
        System.out.printf("%s wins this round! \n", player1.playerName);
        System.out.printf("%s's total score is: %d\n", player1.playerName, player1.playerScore);
        System.out.printf("%s's total score is: %d\n",player2.playerName, player2.playerScore);
        System.out.println("\n");


    } else if(player2.playerScore > player1.playerScore) {
        System.out.printf("%s wins this round! \n", player2.playerName);
        System.out.printf("%s 's total score is: %d\n", player2.playerName, player2.playerScore);
        System.out.printf("%s 's total score is: %d\n",player1.playerName, player1.playerScore);
        System.out.println("\n");
```

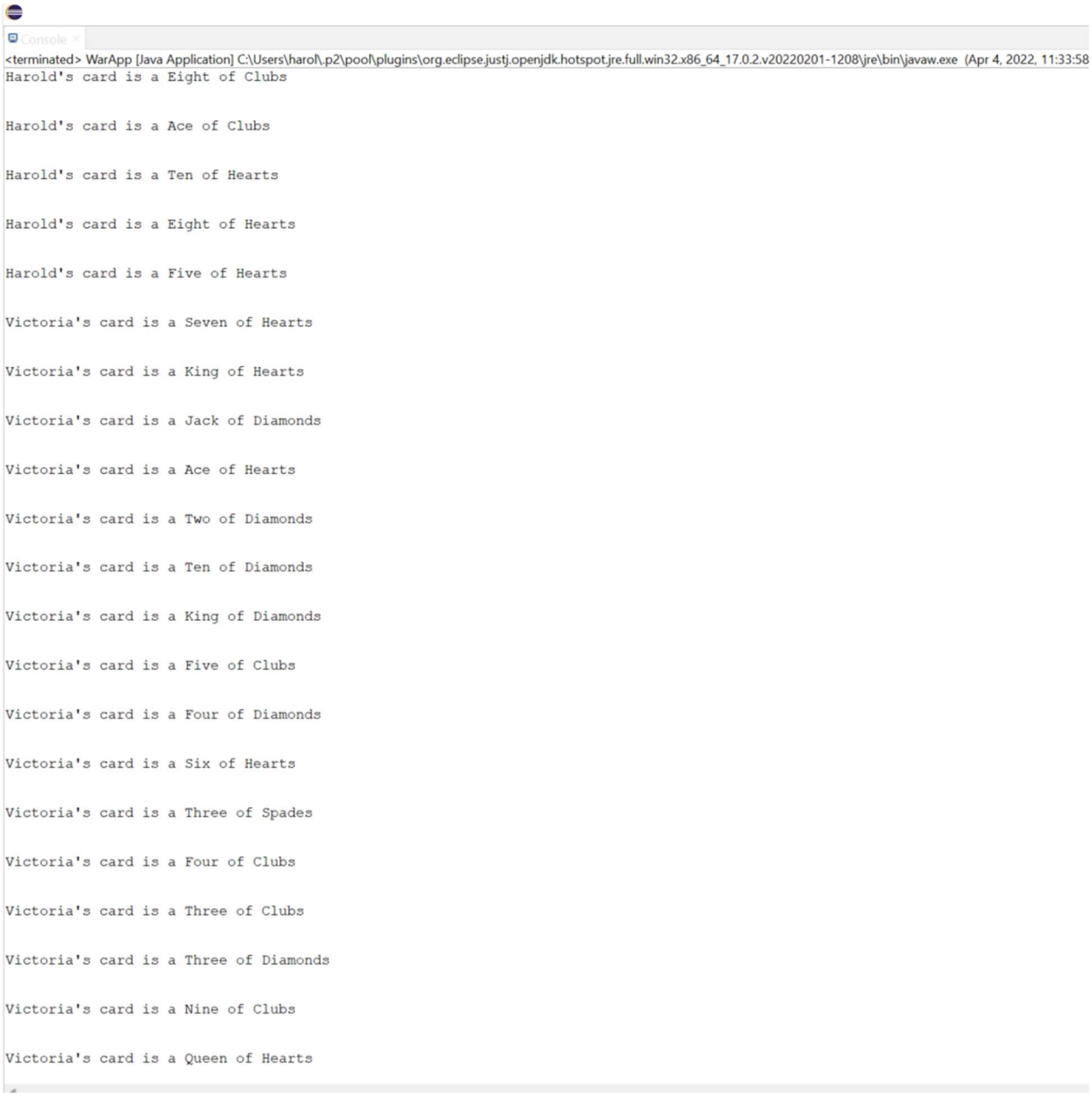
Player.java App.java

```
    Deck newDeck = new Deck();

    newDeck.shuffle();

    Player player1 = new Player("Harold");
    Player player2 = new Player("Victoria");

    // Question 4 -created a for loop to iterate 52 items on the draw method
    for(int i = 0; i < 52; i++) {
      if(i % 2 == 0) {
        player1.hand.add(newDeck.draw());
      }else {
        player2.hand.add(newDeck.draw());
      }
    }

    player1.describe();
    player2.describe();

    // used a for loop to iterate 26 different times calling on the flip method.
    for(int i = 1; i<=26; i++) {
      Card player1Card = player1.flip();
      Card player2Card = player2.flip();
      System.out.println("ROUND" + ": " + i);

      // compared the players scores by calling on the getCardValue method.
      if(player1Card.getCardValue() > player2Card.getCardValue()) {
        player1.incrementScore();

      }else if(player2Card.getCardValue() > player1Card.getCardValue()) {
        player2.incrementScore();

      }else if(player1Card.getCardValue() == player2Card.getCardValue()) {
        player1.incrementScore();
        player2.incrementScore();

      }
      // comparing the scores between calling on the playerScore & playerName methods.
      if (player1.playerScore > player2.playerScore) {
        System.out.printf("%s wins this round! \n", player1.playerName);
        System.out.printf("%s's total score is: %d\n", player1.playerName, player1.playerScore);
        System.out.printf("%s's total score is: %d\n",player2.playerName, player2.playerScore);
        System.out.println("\n");


    } else if(player2.playerScore > player1.playerScore) {
        System.out.printf("%s wins this round! \n", player2.playerName);
        System.out.printf("%s 's total score is: %d\n", player2.playerName, player2.playerScore);
        System.out.printf("%s 's total score is: %d\n",player1.playerName, player1.playerScore);
        System.out.println("\n");
    } else {
        System.err.println("Draw! There are no winners this round!\n");
        System.out.printf("%s 's total score is: %d\n", player1.playerName, player1.playerScore);
        System.out.printf("%s 's total score is: %d\n", player2.playerName, player2.playerScore);
        System.out.println("\n");
   }
  }
 }
}
```

Screenshots of Running Application:




```
Harold's card is a Two of Hearts

Harold's card is a Seven of Spades

Harold's card is a Five of Spades

Harold's card is a Ace of Spades

Harold's card is a Queen of Clubs

Harold's card is a Eight of Spades

Harold's card is a Three of Hearts

Harold's card is a Queen of Diamonds

Harold's card is a Seven of Clubs

Harold's card is a Six of Diamonds

Harold's card is a Jack of Clubs

Harold's card is a Ace of Diamonds

Harold's card is a Jack of Spades

Harold's card is a Six of Spades

Harold's card is a Five of Diamonds

Harold's card is a Four of Spades

Harold's card is a Two of Clubs

Harold's card is a Four of Hearts

Harold's card is a Two of Spades

Harold's card is a Nine of Diamonds

Harold's card is a Six of Clubs
```


Console
<terminated> WarApp [Java Application] C:\Users\harol\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (Apr 4, 2022, 11:33:58
Harold's card is a Eight of Clubs
Harold's card is a Ace of Clubs
Harold's card is a Ten of Hearts
Harold's card is a Eight of Hearts
Harold's card is a Five of Hearts
Victoria's card is a Seven of Hearts
Victoria's card is a King of Hearts
Victoria's card is a Jack of Diamonds
Victoria's card is a Ace of Hearts
Victoria's card is a Two of Diamonds
Victoria's card is a Ten of Diamonds
Victoria's card is a King of Diamonds
Victoria's card is a Five of Clubs
Victoria's card is a Four of Diamonds
Victoria's card is a Six of Hearts
Victoria's card is a Three of Spades
Victoria's card is a Four of Clubs
Victoria's card is a Three of Clubs
Victoria's card is a Three of Diamonds
Victoria's card is a Nine of Clubs
Victoria's card is a Queen of Hearts

Console

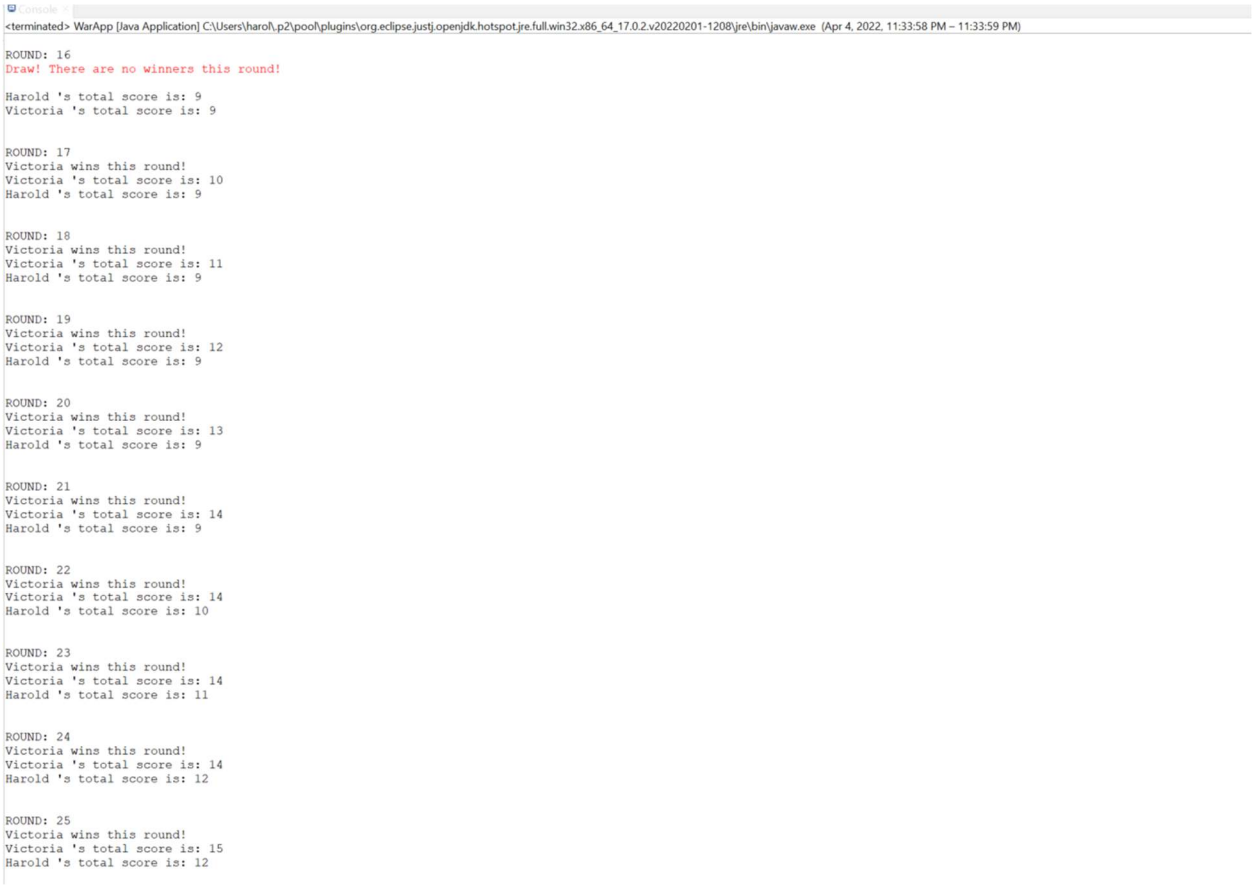
<terminated> WarApp [Java Application] C:\Users\harol\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (Apr 4, 2022, 11:33:58 PM – 11:33:59 PM)

```
Victoria's card is a Queen of Hearts


Victoria's card is a Ten of Spades


Victoria's card is a Nine of Hearts


Victoria's card is a Queen of Spades


Victoria's card is a King of Clubs


Victoria's card is a King of Spades


Victoria's card is a Seven of Diamonds


Victoria's card is a Nine of Spades


Victoria's card is a Eight of Diamonds


Victoria's card is a Ten of Clubs


Victoria's card is a Jack of Hearts


ROUND: 1
Victoria wins this round!
Victoria 's total score is: 1
Harold 's total score is: 0


ROUND: 2
Victoria wins this round!
Victoria 's total score is: 2
Harold 's total score is: 0


ROUND: 3
Victoria wins this round!
Victoria 's total score is: 3
Harold 's total score is: 0


ROUND: 4
Victoria wins this round!
Victoria 's total score is: 4
Harold 's total score is: 1


ROUND: 5
Victoria wins this round!
Victoria 's total score is: 4
Harold 's total score is: 2
```

Console

<terminated> WarApp [Java Application] C:\Users\harol\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (Apr 4, 2022, 11:33:58 PM – 11:33:59 PM)

```
ROUND: 6
Victoria wins this round!
Victoria 's total score is: 5
Harold 's total score is: 2


ROUND: 7
Victoria wins this round!
Victoria 's total score is: 6
Harold 's total score is: 2


ROUND: 8
Victoria wins this round!
Victoria 's total score is: 6
Harold 's total score is: 3


ROUND: 9
Victoria wins this round!
Victoria 's total score is: 6
Harold 's total score is: 4


ROUND: 10
Victoria wins this round!
Victoria 's total score is: 7
Harold 's total score is: 5


ROUND: 11
Victoria wins this round!
Victoria 's total score is: 7
Harold 's total score is: 6


ROUND: 12
HaroldDraw! There are no winners this round!

 's total score is: 7
Victoria 's total score is: 7


ROUND: 13
Harold wins this round!
Harold's total score is: 8
Victoria's total score is: 7


ROUND: 14
Harold wins this round!
Harold's total score is: 9
Victoria's total score is: 7


ROUND: 15
Harold wins this round!
Harold's total score is: 9
Victoria's total score is: 8
```



URL to GitHub Repository:

https://github.com/HaroldLujan/Week-6-Final-Coding-Project.git