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)

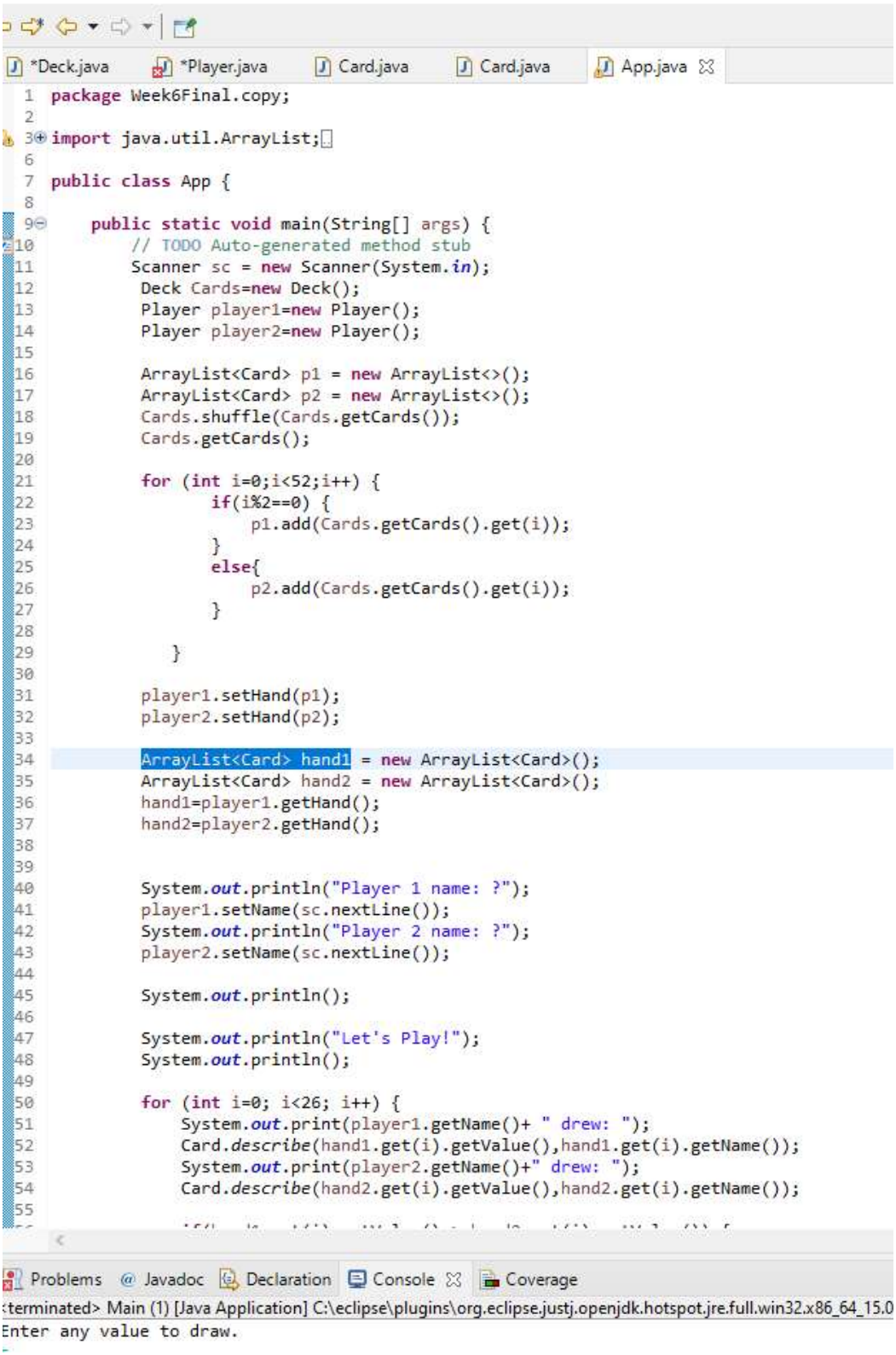
3. In the constructor, when a new Deck is instantiated, the Cards field should be populated with the standard 52 cards.

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:

App*


*Deck.java
*Player.java
Card.java
Card.java
App.java
1 package Week6Final.copy;
2
3 import java.util.ArrayList;
6
7 public class App {
8
9 public static void main(String[] args) {
10 // TODO Auto-generated method stub
11 Scanner sc = new Scanner(System.in);
12 Deck Cards=new Deck();
13 Player player1=new Player();
14 Player player2=new Player();
15
16 ArrayList<Card> p1 = new ArrayList<>();
17 ArrayList<Card> p2 = new ArrayList<>();
18 Cards.shuffle(Cards.getCards());
19 Cards.getCards();
20
21 for (int i=0;i<52;i++) {
22 if(i%2==0) {
23 p1.add(Cards.getCards().get(i));
24 }
25 else{
26 p2.add(Cards.getCards().get(i));
27 }
28
29 }
30
31 player1.setHand(p1);
32 player2.setHand(p2);
33
34 ArrayList<Card> hand1 = new ArrayList<Card>();
35 ArrayList<Card> hand2 = new ArrayList<Card>();
36 hand1=player1.getHand();
37 hand2=player2.getHand();
38
39
40 System.out.println("Player 1 name: ?");
41 player1.setName(sc.nextLine());
42 System.out.println("Player 2 name: ?");
43 player2.setName(sc.nextLine());
44
45 System.out.println();
46
47 System.out.println("Let's Play!");
48 System.out.println();
49
50 for (int i=0; i<26; i++) {
51 System.out.print(player1.getName()+ " drew: ");
52 Card.describe(hand1.get(i).getValue(),hand1.get(i).getName());
53 System.out.print(player2.getName()+" drew: ");
54 Card.describe(hand2.get(i).getValue(),hand2.get(i).getName());
55
Problems
Javadoc
Declaration
Console
Coverage
<terminated> Main (1) [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0
Enter any value to draw.

```
            if(hand1.get(i).getValue() > hand2.get(i).getValue()) {
                player1.setScore(player1.getScore()+1);
                System.out.println(player1.getName()+" won this hand!");
            }
            else if(hand1.get(i).getValue() < hand2.get(i).getValue()) {
                player2.setScore(player2.getScore()+1);
                System.out.println(player2.getName()+" won this hand!");

            }
            else
                System.out.println("This hand was a draw!");

            System.out.println();
            if (i<25) {
                System.out.println("Enter any value to draw.");
                sc.nextLine();
                System.out.println();
            }
        }

        System.out.println(player1.getName()+"'s score: "+player1.getScore());
        System.out.println(player2.getName()+"'s score: "+player2.getScore());

        if (player2.getScore()<player1.getScore()) {
            System.out.println(player1.getName()+" won the match!");
        }
        else if (player2.getScore()>player1.getScore()) {
            System.out.println(player2.getName()+" won the match!");
        }
        else if (player2.getScore()==player1.getScore()) {
            System.out.println("This match ended in a draw!");
        }

        sc.close();


}
```

Card*

*Deck.java *Player.java Card.java App.java

```java
package Week6Final.copy;

import java.util.Arrays;

public class Card {
    private int value;
    private String name;

    public Card(int value, String name) {
        super();
        setValue(value);
        setName(name);
        //describe(getValue(),getName());
    }

    public String toString() {
        return String.format("%s of %s",  getValue(),getName());
    }

    public static void describe(int num, String suit) {
        if (num>=3 && num <= 11)
            System.out.println(num-1 + " of "+ suit);
        else if (num ==2)
            System.out.println("Ace of "+ suit);
        else if (num ==12)
            System.out.println("Jack of "+ suit);
        else if (num ==13)
            System.out.println("Queen of "+ suit);
        else if (num ==14)
            System.out.println("King of "+ suit);

    }


    public int getValue() {
        return value;
    }


    public void setValue(int value) {

        List<Integer> validValues= validValues();
        if(validValues.contains(value)) {
            this.value = value;
        }
    }


    public String getName() {
        return name;
    }


```

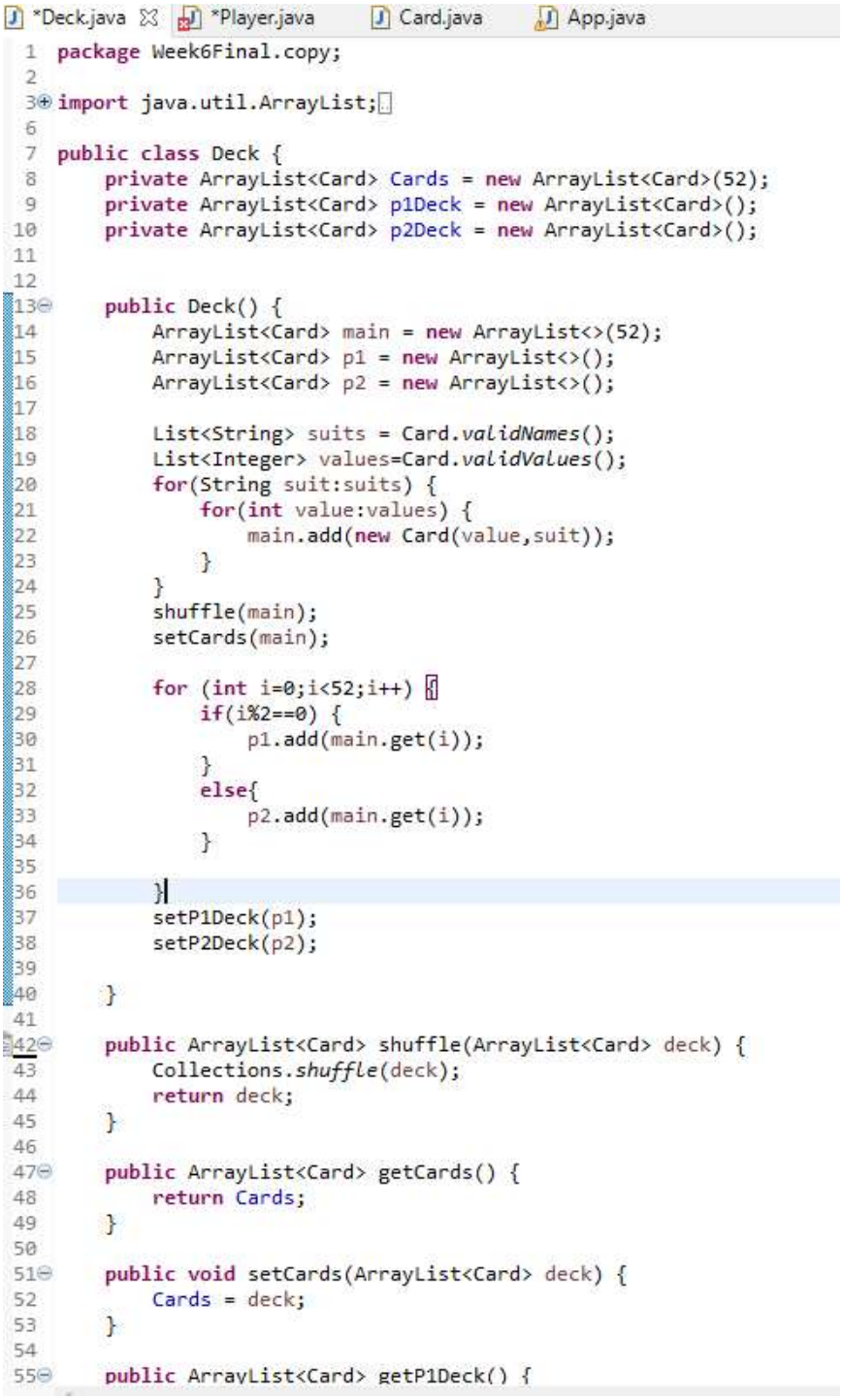
```
54
55      public void setName(String name) {
56          List<String> validNames=validNames();
57          if(validNames.contains(name)) {
58              this.name = name;
59          }
60      }
61
62      public static List<Integer> validValues(){
63          return Arrays.asList(2,3,4,5,6,7,8,9,10,11,12,13,14);
64
65      }
66
67      public static List<String> validNames(){
68          return Arrays.asList("diamonds","clubs","spades","hearts");
69      }
70  }
71
72
```

Player*

*Deck.java | *Player.java | Card.java | App.java

```
1  package Week6Final.copy;
2
3  import java.util.ArrayList;
4
5  public class Player {
6      private ArrayList<Card> hand;
7      private int score;
8      private String name;
9
10
11     public Player() {
12         super();
13         this.hand = hand;
14         this.score = 0;
15         this.name = name;
16     }
17
18     public static void describe (ArrayList<Card> hand) {
19         System.out.println(hand);
20     }
21
22     public Card flip (ArrayList<Card> hand) {
23         Card card=hand.get(0);
24         hand.remove(0);
25         setHand(hand);
26         return card;
27
28     }
29
30     public ArrayList<Card> getHand() {
31         return hand;
32     }
33
34     public void setHand(ArrayList<Card> hand) {
35         this.hand = hand;
36     }
37
38     public int getScore() {
39         return score;
40     }
41
42     public void setScore(int score) {
43         this.score = score;
44     }
45
46     public String getName() {
47         return name;
48     }
49
50     public void setName(String name) {
51         this.name = name;
52     }
53
```

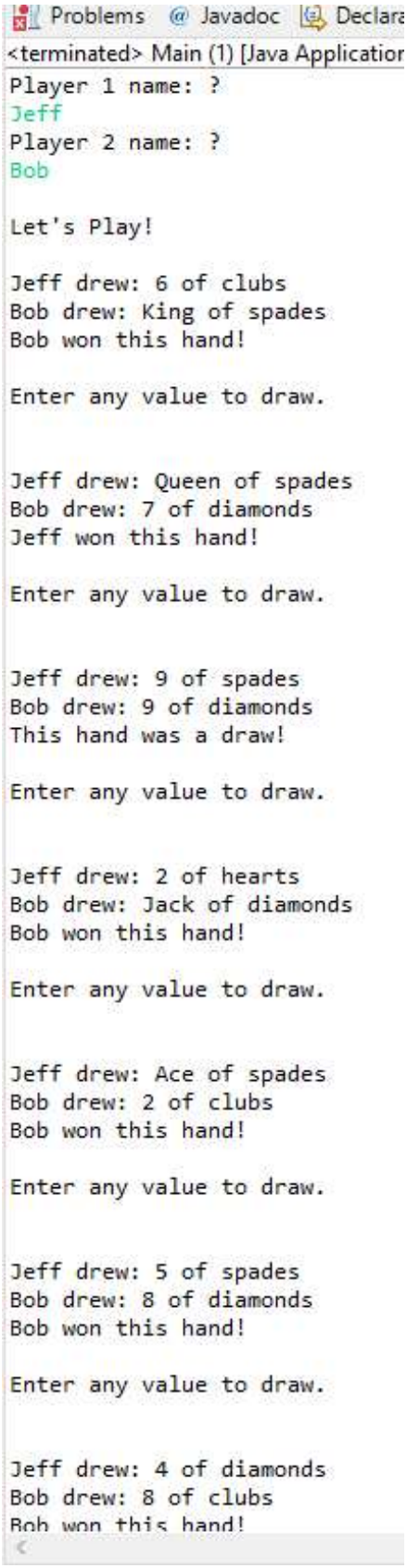
Deck*

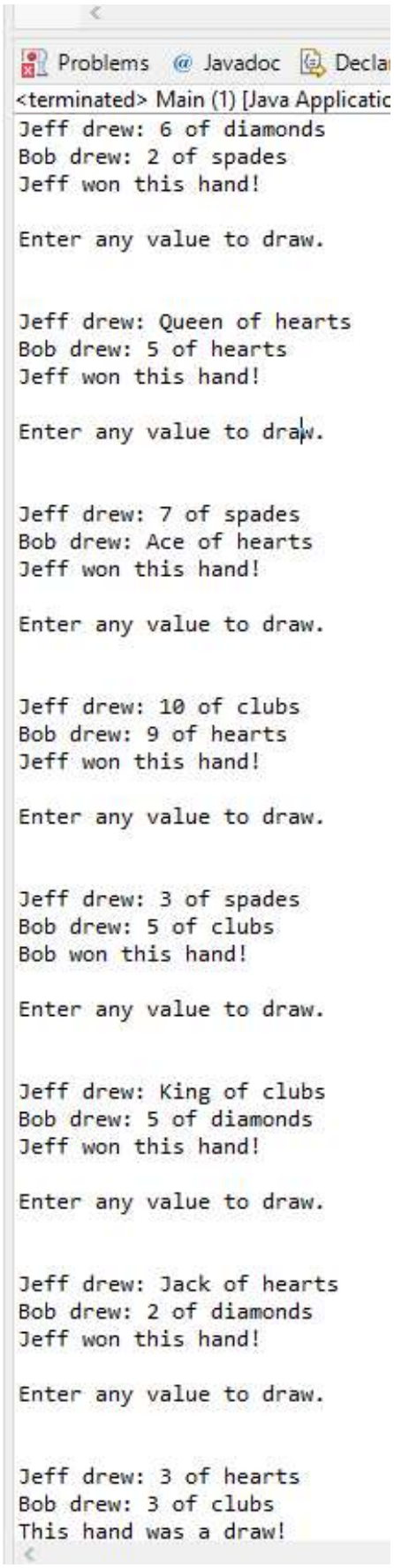



```java
package Week6Final.copy;

import java.util.ArrayList;

public class Deck {
    private ArrayList<Card> Cards = new ArrayList<Card>(52);
    private ArrayList<Card> p1Deck = new ArrayList<Card>();
    private ArrayList<Card> p2Deck = new ArrayList<Card>();


    public Deck() {
        ArrayList<Card> main = new ArrayList<>(52);
        ArrayList<Card> p1 = new ArrayList<>();
        ArrayList<Card> p2 = new ArrayList<>();

        List<String> suits = Card.validNames();
        List<Integer> values=Card.validValues();
        for(String suit:suits) {
            for(int value:values) {
                main.add(new Card(value,suit));
            }
        }
        shuffle(main);
        setCards(main);

        for (int i=0;i<52;i++) {
            if(i%2==0) {
                p1.add(main.get(i));
            }
            else{
                p2.add(main.get(i));
            }

        }
        setP1Deck(p1);
        setP2Deck(p2);

    }

    public ArrayList<Card> shuffle(ArrayList<Card> deck) {
        Collections.shuffle(deck);
        return deck;
    }

    public ArrayList<Card> getCards() {
        return Cards;
    }

    public void setCards(ArrayList<Card> deck) {
        Cards = deck;
    }

    public ArrayList<Card> getP1Deck() {
```

```
55      public ArrayList<Card> getP1Deck() {
56          return p1Deck;
57      }
58
59      public void setP1Deck(ArrayList<Card> p1Deck) {
60          this.p1Deck = p1Deck;
61      }
62
63      public ArrayList<Card> getP2Deck() {
64          return p2Deck;
65      }
66
67      public void setP2Deck(ArrayList<Card> p2Deck) {
68          this.p2Deck = p2Deck;
69      }
70
71
72  }
73
74
```

Screenshots of Running Application:


Problems
Javadoc
Declara
<terminated> Main (1) [Java Applicatior
Player 1 name: ?
Jeff
Player 2 name: ?
Bob
Let's Play!
Jeff drew: 6 of clubs
Bob drew: King of spades
Bob won this hand!
Enter any value to draw.
Jeff drew: Queen of spades
Bob drew: 7 of diamonds
Jeff won this hand!
Enter any value to draw.
Jeff drew: 9 of spades
Bob drew: 9 of diamonds
This hand was a draw!
Enter any value to draw.
Jeff drew: 2 of hearts
Bob drew: Jack of diamonds
Bob won this hand!
Enter any value to draw.
Jeff drew: Ace of spades
Bob drew: 2 of clubs
Bob won this hand!
Enter any value to draw.
Jeff drew: 5 of spades
Bob drew: 8 of diamonds
Bob won this hand!
Enter any value to draw.
Jeff drew: 4 of diamonds
Bob drew: 8 of clubs
Bob won this hand!


Problems
Javadoc
Decla
<terminated> Main (1) [Java Applicatic
Jeff drew: 6 of diamonds
Bob drew: 2 of spades
Jeff won this hand!
Enter any value to draw.
Jeff drew: Queen of hearts
Bob drew: 5 of hearts
Jeff won this hand!
Enter any value to draw.
Jeff drew: 7 of spades
Bob drew: Ace of hearts
Jeff won this hand!
Enter any value to draw.
Jeff drew: 10 of clubs
Bob drew: 9 of hearts
Jeff won this hand!
Enter any value to draw.
Jeff drew: 3 of spades
Bob drew: 5 of clubs
Bob won this hand!
Enter any value to draw.
Jeff drew: King of clubs
Bob drew: 5 of diamonds
Jeff won this hand!
Enter any value to draw.
Jeff drew: Jack of hearts
Bob drew: 2 of diamonds
Jeff won this hand!
Enter any value to draw.
Jeff drew: 3 of hearts
Bob drew: 3 of clubs
This hand was a draw!


```
Enter any value to draw.


Jeff drew: 8 of hearts
Bob drew: Jack of clubs
Bob won this hand!

Enter any value to draw.


Jeff drew: 9 of clubs
Bob drew: 4 of hearts
Jeff won this hand!

Enter any value to draw.


Jeff drew: 3 of diamonds
Bob drew: King of hearts
Bob won this hand!

Enter any value to draw.


Jeff drew: 6 of spades
Bob drew: Ace of diamonds
Jeff won this hand!

Enter any value to draw.


Jeff drew: Queen of clubs
Bob drew: 7 of clubs
Jeff won this hand!

Enter any value to draw.


Jeff drew: King of diamonds
Bob drew: 4 of clubs
Jeff won this hand!

Enter any value to draw.


Jeff drew: Ace of clubs
Bob drew: 10 of spades
Bob won this hand!

Enter any value to draw.
```


```
Jeff drew: Queen of clubs
Bob drew: 7 of clubs
Jeff won this hand!

Enter any value to draw.


Jeff drew: King of diamonds
Bob drew: 4 of clubs
Jeff won this hand!

Enter any value to draw.


Jeff drew: Ace of clubs
Bob drew: 10 of spades
Bob won this hand!

Enter any value to draw.


Jeff drew: 6 of hearts
Bob drew: 8 of spades
Bob won this hand!

Enter any value to draw.


Jeff drew: Jack of spades
Bob drew: 4 of spades
Jeff won this hand!

Enter any value to draw.


Jeff drew: 10 of diamonds
Bob drew: Queen of diamonds
Bob won this hand!

Enter any value to draw.


Jeff drew: 10 of hearts
Bob drew: 7 of hearts
Jeff won this hand!

Jeff's score: 13
Bob's score: 11
Jeff won the match!
```

URL to GitHub Repository:

https://github.com/marlon214/Week6Final