

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:

The screenshot shows an IDE interface with several tabs: App.java, Card.java, Deck.java, Player.java, and Console. The App.java tab is active, displaying the following Java code:

```
App.java
1 package com.lisasmith.week6;
2
3 public class App {
4
5     public static void main(String[] args) {
6
7
8         /*
9          * IntroToJava Week 6 Coding Assignment
10         * Create an automated version of the
11         * classic card game WAR!
12         */
13
14     System.out.println("-----");
15     System.out.println(" Intro to Java Week 6 Coding Assignment");
16     System.out.println("-----");
17
18     System.out.println();
19
20     System.out.println("-----");
21     System.out.println(" Start: Game of WAR");
22     System.out.println("-----");
23     System.out.println();
24
25
26
27
28
29
30
31     /*
32      * Requirement 3: Instantiate a Deck and two players
33      */
34
35     Deck deck = new Deck();
36     Player playerOne = new Player("Bobby");
37     Player playerTwo = new Player("Sally");
38
39     System.out.println("Player 1: " + playerOne.getName() + " VS. Player 2: " + playerTwo.getName());
40
41
42     /*
43      * Requirement 3: Shuffle the Deck of 52 Cards
44      */
45
46     System.out.println(" Deck is shuffled");
47     System.out.println();
48     System.out.println();
49     deck.shuffle();
50
51
52     /*
53      * Requirement 4: Using a traditional for loop, iterate 52 times
54      * calling the draw method, alternating between playerOne and playerTwo
55      * using the instantiated Deck.
56      */
57
58     System.out.println(" Players draw their hands...");
59     System.out.println();
60     System.out.println();
61     for (int i=0; i<52; i++) {
62         if (i % 2 == 1) {
63             playerOne.draw(deck);
64         } else {
65             playerTwo.draw(deck);
66         }
67     }
68
69
70     /*
71      * Requirement 5: Using a traditional for loop, iterate 26 times,
72      * and call the flip method for each player.
73      */
74
75
76     System.out.println(" Starting to play....");
77     System.out.println();
78     System.out.println();
79     Card playerOneTopCard = new Card();
80     Card playerTwoTopCard = new Card();
81     for (int i=0; i<26; i++) {
82
83         System.out.println("-----");
84         System.out.println("Turn #: " + (i+1));
85         System.out.println("-----");
86         playerOneTopCard = playerOne.flip(playerOne.hand);
87         System.out.print("Player 1: " + playerOne.getName() + "s ");
88         playerOneTopCard.describe();
89
90
91         playerTwoTopCard = playerTwo.flip(playerTwo.hand);
92         System.out.print("Player 2: " + playerTwo.getName() + "s ");
93         playerTwoTopCard.describe();
94
95
96
97         playerOneTopCard.describe();
98
99         playerTwoTopCard = playerTwo.flip(playerTwo.hand);
100        System.out.print("Player 2: " + playerTwo.getName() + "s ");
101        playerTwoTopCard.describe();
102
103        compareCardAndIncrementScore(playerOne, playerOneTopCard.getValue(), playerTwo, playerTwoTopCard.getValue());
104
105    }
106
107
108    System.out.println("-----");
109
110    /*
111     * Method to find out which player has a higher score:
112     * Value: -1 (playerOne wins)
113     * 0 (playerOne == playerTwo)
114     * 1 (playerTwo wins)
115     */
116
117    int winner = compareFinalScore(playerOne, playerTwo);
118
119    System.out.println("Final Scores:");
120    System.out.println(" Player 1: " + playerOne.getName() + "s score: " + playerOne.getScore());
121    System.out.println(" Player 2: " + playerTwo.getName() + "s score: " + playerTwo.getScore());
122
123    System.out.println();
124    System.out.print(" The Winner of WAR is: ");
125
126    switch (winner) {
127        case (-1):
128            System.out.println("Player 1");
129            break;
130        case (0):
131            System.out.println("It's a DRAW!");
132            break;
133        case (1):
134            System.out.println("Player 2, " + playerTwo.getName());
135            break;
136    }
137
138    System.out.println("-----");
139    System.out.println("----- End: Game of WAR!");
140
141
142    private static int compareFinalScore(Player playerOne, Player playerTwo) {
143
144        int scorePlayerOne = playerOne.getScore();
145        int scorePlayerTwo = playerTwo.getScore();
146
147        if (scorePlayerOne > scorePlayerTwo) {
148            return 1;
149        } else if (scorePlayerOne < scorePlayerTwo) {
150            return -1;
151        } else {
152            return 0;
153        }
154    }
155
156    private static void compareCardAndIncrementScore(Player p1, int value1, Player p2, int value2) {
157
158        System.out.println(" A point for Player 1, " + p1.getName());
159        System.out.println(" A point for Player 2, " + p2.getName());
160
161        if (value1 == value2) {
162            System.out.println(" The values are equal, no score!");
163        }
164
165    }
166
167
168}
```

```
App.java Card.java Deck.java Player.java Console
```

```
1 package com.lisasmith.week6;
2
3 public class Card {
4
5     // contains a value from 2-14 representing cards 2-Ace
6     int value;
7     // (e.g. "Ace of Diamonds", or "Two of Hearts")
8     String name;
9
10
11    public int getValue() {
12        return this.value;
13    }
14
15
16
17    public void setValue(int value) {
18        this.value = value;
19    }
20
21
22
23    public String getName() {
24        return this.name;
25    }
26
27
28
29    public void setName(String name) {
30        this.name = name;
31    }
32
33
34    /*
35     * The method describe() prints out information about a card
36     */
37    public void describe() {
38        System.out.println("Card is: " + this.getName());
39        System.out.println("Value is: " + this.getValue());
40        System.out.println();
41    }
42
43 }
44
45 }
```

```
App.java Card.java
```

```
1 package com.lisasmith.week6;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class Card {
7     List<Card> hand = new ArrayList<Card>();
8     int score;
9     String name;
10
11    /*
12     * Constructor
13     */
14    public Card(String name) {
15        this.name = name;
16    }
17
18
19    /*
20     * Getters & Setters
21     */
22
23    public List<Card> getHand() {
24        return hand;
25    }
26
27
28
29    public void setHand(List<Card> hand) {
30        this.hand = hand;
31    }
32
33
34    public int getScore() {
35        return score;
36    }
37
38
39    public void setScore(int score) {
40        this.score = score;
41    }
42
43
44
45    public String getName() {
46        return name;
47    }
48
49 }
```

App.java Card.java Deck.java Player.java Console

```
1 package com.lisasmith.week6;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class Player {
7     private List<Card> hand = new ArrayList<Card>();
8     int score;
9     String name;
10
11     /*
12      * Constructor
13      */
14     public Player(String name) {
15         this.score = 0;
16         this.name = name;
17     }
18
19     /*
20      * Getters & Setters
21      */
22     public List<Card> getHand() {
23         return hand;
24     }
25
26     public void setHand(List<Card> hand) {
27         this.hand = hand;
28     }
29
30     public int getScore() {
31         return score;
32     }
33
34     public void setScore(int score) {
35         this.score = score;
36     }
37
38     public String getName() {
39         return name;
40     }
41
42     public void setName(String name) {
43         this.name = name;
44     }
45
46     /*
47      * Prints out information about the player and calls the
48      * describe() method for each card in the Hand List
49      */
50     public void describe() {
51         System.out.println("Player's Name: " + this.name);
52         System.out.println("Player's Score: " + this.score);
53         System.out.println("Player's Hand:");
54         int count = 1;
55         for (Card handCard : this.hand) {
56             System.out.print ("Card" + count + ":");
57             handCard.describe();
58             count++;
59         }
60         System.out.println();
61     }
62
63     /*
64      * Removes and returns the top card of the hand
65      */
66     public Card flip(List<Card> hand) {
67         Card topCard = new Card();
68         topCard = hand.get(0);
69         hand.remove(0);
70         return topCard;
71     }
72
73     /*
74      * Takes a Deck as an argument and calls the draw method
75      * on the deck, adding the returned Card to the hand field.
76      */
77     public void draw(Deck deck) {
78         Card newCard = new Card();
79
80         if ((newCard = deck.draw(deck.cards)) != null) {
81             this.hand.add(newCard);
82         } else {
83             System.out.println("    Deck Empty");
84         }
85     }
86
87     /*
88      * Adds 1 to the Player's score field
89      */
90     public void incrementScore () {
91         this.score++;
92     }
93
94 }
```

App.java

```
1 package com.lisasmith.week6;
2
3 import java.util.ArrayList;
4 import java.util.Collections;
5 import java.util.List;
6
7 public class Deck {
8
9     List<Card> cards = new ArrayList<Card>();
10
11     /*
12      * Getters & Setters
13      */
14
15     public List<Card> getCards() {
16         return cards;
17     }
18
19     public void setCards(List<Card> cards) {
20         this.cards = cards;
21     }
22
23     /*
24      * Constructor:
25      */
26     public Deck() {
27         populateDeck(cards);
28     }
29
30     /*
31      * Randomizes the order of the cards
32      */
33     public void shuffle() {
34         Collections.shuffle(cards);
35     }
36
37     /*
38      * Removes and returns the top card of the Cards field
39      */
40     public Card draw(List<Card> cards) {
41         Card topCard = new Card();
42         if (cards.size() == 0) {
43             System.out.println("No cards left == deck empty!");
44             return null;
45         } else {
46             topCard = cards.get(0);
47             cards.remove(0);
48             return topCard;
49         }
50     }
51
52     private List<Card> createFourSuits(int value, String num) {
53         List<Card> newList = new ArrayList<Card>();
54
55         for (int i = 0; i < 4; i++) {
56             Card newCard = new Card();
57             newCard.setValue(value);
58             switch (i) {
59                 case (0):
60                     newCard.setName(num + " of Hearts");
61                     newList.add(newCard);
62                     break;
63                 case (1):
64                     newCard.setName(num + " of Diamonds");
65                     newList.add(newCard);
66                     break;
67                 case (2):
68                     newCard.setName(num + " of Clubs");
69                     newList.add(newCard);
70                     break;
71                 case (3):
72                     newCard.setName(num + " of Spades");
73                     newList.add(newCard);
74                     break;
75             }
76         }
77         return newList;
78     }
79
80     private void populateDeck(List<Card> cards) {
81         for (int i = 2; i <= 14; i++) {
82             switch (i) {
83                 case (2):
84                     cards.addAll(createFourSuits(i, "Two"));
85                     break;
86                 case (3):
87                     cards.addAll(createFourSuits(i, "Three"));
88                     break;
89                 case (4):
90                     cards.addAll(createFourSuits(i, "Four"));
91                     break;
92                 case (5):
93                     cards.addAll(createFourSuits(i, "Five"));
94                     break;
95                 case (6):
96                     cards.addAll(createFourSuits(i, "Six"));
97                     break;
98                 case (7):
99                     cards.addAll(createFourSuits(i, "Seven"));
100                    break;
101                 case (8):
102                     cards.addAll(createFourSuits(i, "Eight"));
103                     break;
104                 case (9):
105                     cards.addAll(createFourSuits(i, "Nine"));
106                     break;
107                 case (10):
108                     cards.addAll(createFourSuits(i, "Ten"));
109                     break;
110                 case (11):
111                     cards.addAll(createFourSuits(i, "Jack"));
112                     break;
113                 case (12):
114                     cards.addAll(createFourSuits(i, "Queen"));
115                     break;
116                 case (13):
117                     cards.addAll(createFourSuits(i, "King"));
118                     break;
119                 case (14):
120                     cards.addAll(createFourSuits(i, "Ace"));
121                     break;
122             }
123         }
124     }
125 }
126 }
```

Card.java

```
1 package com.lisasmith.week6;
2
3 public class Card {
4
5     int value;
6
7     String name;
8
9     public int getValue() {
10        return value;
11    }
12
13    public void setValue(int value) {
14        this.value = value;
15    }
16
17    public String getName() {
18        return name;
19    }
20
21    public void setName(String name) {
22        this.name = name;
23    }
24 }
```

*Deck.java

```
1 package com.lisasmith.week6;
2
3 public class Deck {
4
5     List<Card> cards = new ArrayList<Card>();
6
7     /*
8      * Getters & Setters
9      */
10
11     public List<Card> getCards() {
12         return cards;
13     }
14
15     public void setCards(List<Card> cards) {
16         this.cards = cards;
17     }
18
19     /*
20      * Randomizes the order of the cards
21      */
22     public void shuffle() {
23         Collections.shuffle(cards);
24     }
25
26     /*
27      * Removes and returns the top card of the Cards field
28      */
29     public Card draw(List<Card> cards) {
30         Card topCard = new Card();
31         if (cards.size() == 0) {
32             System.out.println("No cards left == deck empty!");
33             return null;
34         } else {
35             topCard = cards.get(0);
36             cards.remove(0);
37             return topCard;
38         }
39     }
40
41     private List<Card> createFourSuits(int value, String num) {
42         List<Card> newList = new ArrayList<Card>();
43
44         for (int i = 0; i < 4; i++) {
45             Card newCard = new Card();
46             newCard.setValue(value);
47             switch (i) {
48                 case (0):
49                     newCard.setName(num + " of Hearts");
50                     newList.add(newCard);
51                     break;
52                 case (1):
53                     newCard.setName(num + " of Diamonds");
54                     newList.add(newCard);
55                     break;
56                 case (2):
57                     newCard.setName(num + " of Clubs");
58                     newList.add(newCard);
59                     break;
60                 case (3):
61                     newCard.setName(num + " of Spades");
62                     newList.add(newCard);
63                     break;
64             }
65         }
66         return newList;
67     }
68
69     private void populateDeck(List<Card> cards) {
70         for (int i = 2; i <= 14; i++) {
71             switch (i) {
72                 case (2):
73                     cards.addAll(createFourSuits(i, "Two"));
74                     break;
75                 case (3):
76                     cards.addAll(createFourSuits(i, "Three"));
77                     break;
78                 case (4):
79                     cards.addAll(createFourSuits(i, "Four"));
80                     break;
81                 case (5):
82                     cards.addAll(createFourSuits(i, "Five"));
83                     break;
84                 case (6):
85                     cards.addAll(createFourSuits(i, "Six"));
86                     break;
87                 case (7):
88                     cards.addAll(createFourSuits(i, "Seven"));
89                     break;
90                 case (8):
91                     cards.addAll(createFourSuits(i, "Eight"));
92                     break;
93                 case (9):
94                     cards.addAll(createFourSuits(i, "Nine"));
95                     break;
96                 case (10):
97                     cards.addAll(createFourSuits(i, "Ten"));
98                     break;
99                 case (11):
100                    cards.addAll(createFourSuits(i, "Jack"));
101                    break;
102                 case (12):
103                     cards.addAll(createFourSuits(i, "Queen"));
104                     break;
105                 case (13):
106                     cards.addAll(createFourSuits(i, "King"));
107                     break;
108                 case (14):
109                     cards.addAll(createFourSuits(i, "Ace"));
110                     break;
111             }
112         }
113     }
114 }
```

Player.java

Screenshots of Running Application:
Run #1

```
<terminated> App (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_261.jdk/ContentRoot/bin/Intro to Java Week 6 Coding Assignment
-----
Start: Game of WAR
-----
Player 1, Bobby VS. Player 2, Sally
Deck is shuffled
Players draw their hands...
Starting to play....
-----
Turn #: 1
Player 1, Bobby's Card is: Two of Spades
Value is: 2
Player 2, Sally's Card is: Ace of Diamonds
Value is: 14
A point for Player 2, Sally
-----
Turn #: 2
Player 1, Bobby's Card is: Eight of Hearts
Value is: 8
Player 2, Sally's Card is: King of Diamonds
Value is: 13
A point for Player 2, Sally
-----
Turn #: 3
Player 1, Bobby's Card is: Nine of Hearts
Value is: 9
Player 2, Sally's Card is: Two of Clubs
Value is: 2
A point to Player 1, Bobby
-----
Turn #: 4
Player 1, Bobby's Card is: King of Spades
Value is: 13
Player 2, Sally's Card is: Five of Spades
Value is: 5
A point to Player 1, Bobby
-----
Turn #: 5
Player 1, Bobby's Card is: Five of Clubs
Value is: 5
Player 2, Sally's Card is: Six of Spades
Value is: 6
A point for Player 2, Sally
-----
Turn #: 6
Player 1, Bobby's Card is: Nine of Spades
Value is: 9
Player 2, Sally's Card is: Jack of Diamonds
Value is: 11
A point for Player 2, Sally
-----
Turn #: 7
Player 1, Bobby's Card is: Four of Diamonds
Value is: 4
Player 2, Sally's Card is: Five of Hearts
Value is: 5
A point for Player 2, Sally
-----
Turn #: 8
Player 1, Bobby's Card is: Nine of Diamonds
Value is: 9
Player 2, Sally's Card is: Six of Hearts
Value is: 6
A point to Player 1, Bobby
-----
Turn #: 9
Player 1, Bobby's Card is: Six of Diamonds
Value is: 6
Player 2, Sally's Card is: Three of Hearts
Value is: 3
A point to Player 1, Bobby
-----
Turn #: 10
Player 1, Bobby's Card is: Three of Spades
Value is: 3
Player 2, Sally's Card is: Seven of Clubs
Value is: 7
A point for Player 2, Sally
```

Run #2

```
<terminated> App (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_261.jdk/ContentRoot/bin/Intro to Java Week 6 Coding Assignment
-----
Start: Game of WAR
-----
Player 1, Bobby VS. Player 2, Sally
Deck is shuffled
Players draw their hands...
Starting to play....
-----
Turn #: 1
Player 1, Bobby's Card is: Four of Diamonds
Value is: 4
Player 2, Sally's Card is: Five of Diamonds
Value is: 5
A point for Player 2, Sally
-----
Turn #: 2
Player 1, Bobby's Card is: Ten of Diamonds
Value is: 10
Player 2, Sally's Card is: Three of Hearts
Value is: 3
A point to Player 1, Bobby
-----
Turn #: 3
Player 1, Bobby's Card is: Jack of Diamonds
Value is: 11
Player 2, Sally's Card is: Ace of Diamonds
Value is: 14
A point for Player 2, Sally
-----
Turn #: 4
Player 1, Bobby's Card is: Ace of Hearts
Value is: 14
Player 2, Sally's Card is: Five of Hearts
Value is: 5
A point to Player 1, Bobby
-----
Turn #: 5
Player 1, Bobby's Card is: King of Hearts
Value is: 13
Player 2, Sally's Card is: Ten of Clubs
Value is: 10
A point to Player 1, Bobby
-----
Turn #: 6
Player 1, Bobby's Card is: Jack of Hearts
Value is: 11
Player 2, Sally's Card is: Four of Spades
Value is: 4
A point to Player 1, Bobby
-----
Turn #: 7
Player 1, Bobby's Card is: Ace of Clubs
Value is: 14
Player 2, Sally's Card is: Ten of Hearts
Value is: 10
A point to Player 1, Bobby
-----
Turn #: 8
Player 1, Bobby's Card is: Four of Hearts
Value is: 4
Player 2, Sally's Card is: Six of Diamonds
Value is: 6
A point for Player 2, Sally
-----
Turn #: 9
Player 1, Bobby's Card is: Three of Clubs
Value is: 3
Player 2, Sally's Card is: Nine of Diamonds
Value is: 9
A point for Player 2, Sally
-----
Turn #: 10
Player 1, Bobby's Card is: Jack of Spades
Value is: 11
Player 2, Sally's Card is: Nine of Hearts
Value is: 9
A point to Player 1, Bobby
```

(3 distinct runs — side by side!)

Run #3

```
<terminated> App (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_261.jdk/ContentRoot/bin/Intro to Java Week 6 Coding Assignment
-----
Start: Game of WAR
-----
Player 1, Bobby VS. Player 2, Sally
Deck is shuffled
Players draw their hands...
Starting to play....
-----
Turn #: 1
Player 1, Bobby's Card is: Ace of Hearts
Value is: 14
Player 2, Sally's Card is: Five of Diamonds
Value is: 5
A point to Player 1, Bobby
-----
Turn #: 2
Player 1, Bobby's Card is: Six of Hearts
Value is: 6
Player 2, Sally's Card is: Ace of Spades
Value is: 14
A point for Player 2, Sally
-----
Turn #: 3
Player 1, Bobby's Card is: Ace of Clubs
Value is: 14
Player 2, Sally's Card is: Nine of Clubs
Value is: 9
A point to Player 1, Bobby
-----
Turn #: 4
Player 1, Bobby's Card is: King of Clubs
Value is: 13
Player 2, Sally's Card is: King of Hearts
Value is: 13
The values are equal, no score!
-----
Turn #: 5
Player 1, Bobby's Card is: Five of Spades
Value is: 5
Player 2, Sally's Card is: Eight of Clubs
Value is: 8
A point for Player 2, Sally
-----
Turn #: 6
Player 1, Bobby's Card is: Seven of Clubs
Value is: 7
Player 2, Sally's Card is: Jack of Diamonds
Value is: 11
A point for Player 2, Sally
-----
Turn #: 7
Player 1, Bobby's Card is: Two of Spades
Value is: 2
Player 2, Sally's Card is: Jack of Clubs
Value is: 11
A point for Player 2, Sally
-----
Turn #: 8
Player 1, Bobby's Card is: Four of Hearts
Value is: 4
Player 2, Sally's Card is: Nine of Diamonds
Value is: 9
A point for Player 2, Sally
-----
Turn #: 9
Player 1, Bobby's Card is: Eight of Diamonds
Value is: 8
Player 2, Sally's Card is: Jack of Hearts
Value is: 11
A point for Player 2, Sally
-----
Turn #: 10
Player 1, Bobby's Card is: Six of Spades
Value is: 6
Player 2, Sally's Card is: Eight of Hearts
Value is: 8
A point for Player 2, Sally
```

Turn #: 11
Player 1, Bobby's Card is: Queen of Spades
Value is: 12
Player 2, Sally's Card is: Nine of Clubs
Value is: 9
A point to Player 1, Bobby

Turn #: 12
Player 1, Bobby's Card is: Queen of Clubs
Value is: 12
Player 2, Sally's Card is: Three of Clubs
Value is: 3
A point to Player 1, Bobby

Turn #: 13
Player 1, Bobby's Card is: Ace of Clubs
Value is: 14
Player 2, Sally's Card is: Two of Diamonds
Value is: 2
A point to Player 1, Bobby

Turn #: 14
Player 1, Bobby's Card is: King of Clubs
Value is: 13
Player 2, Sally's Card is: Ten of Hearts
Value is: 10
A point to Player 1, Bobby

Turn #: 15
Player 1, Bobby's Card is: Ten of Clubs
Value is: 10
Player 2, Sally's Card is: Four of Spades
Value is: 4
A point to Player 1, Bobby

Turn #: 16
Player 1, Bobby's Card is: Queen of Diamonds
Value is: 12
Player 2, Sally's Card is: Four of Hearts
Value is: 4
A point to Player 1, Bobby

Turn #: 17
Player 1, Bobby's Card is: Two of Hearts
Value is: 2
Player 2, Sally's Card is: Ace of Spades
Value is: 14
A point for Player 2, Sally

Turn #: 18
Player 1, Bobby's Card is: Eight of Diamonds
Value is: 8
Player 2, Sally's Card is: Ten of Diamonds
Value is: 10
A point for Player 2, Sally

Turn #: 19
Player 1, Bobby's Card is: King of Hearts
Value is: 13
Player 2, Sally's Card is: Six of Clubs
Value is: 6
A point to Player 1, Bobby

Turn #: 20
Player 1, Bobby's Card is: Eight of Clubs
Value is: 8
Player 2, Sally's Card is: Three of Diamonds
Value is: 3
A point to Player 1, Bobby

Turn #: 21
Player 1, Bobby's Card is: Queen of Hearts
Value is: 12
Player 2, Sally's Card is: Jack of Clubs
Value is: 11
A point to Player 1, Bobby

Turn #: 22
Player 1, Bobby's Card is: Jack of Spades
Value is: 11
Player 2, Sally's Card is: Seven of Hearts
Value is: 7
A point to Player 1, Bobby

Turn #: 11
Player 1, Bobby's Card is: Seven of Clubs
Value is: 7
Player 2, Sally's Card is: Queen of Diamonds
Value is: 12
A point for Player 2, Sally

Turn #: 12
Player 1, Bobby's Card is: Seven of Hearts
Value is: 7
Player 2, Sally's Card is: Two of Clubs
Value is: 2
A point to Player 1, Bobby

Turn #: 13
Player 1, Bobby's Card is: Six of Spades
Value is: 6
Player 2, Sally's Card is: Eight of Clubs
Value is: 8
A point for Player 2, Sally

Turn #: 14
Player 1, Bobby's Card is: Ten of Spades
Value is: 10
Player 2, Sally's Card is: King of Spades
Value is: 13
A point for Player 2, Sally

Turn #: 15
Player 1, Bobby's Card is: Queen of Spades
Value is: 12
Player 2, Sally's Card is: Seven of Diamonds
Value is: 7
A point to Player 1, Bobby

Turn #: 16
Player 1, Bobby's Card is: Three of Diamonds
Value is: 3
Player 2, Sally's Card is: Two of Diamonds
Value is: 2
A point to Player 1, Bobby

Turn #: 17
Player 1, Bobby's Card is: Five of Clubs
Value is: 5
Player 2, Sally's Card is: Queen of Clubs
Value is: 12
A point for Player 2, Sally

Turn #: 18
Player 1, Bobby's Card is: Nine of Clubs
Value is: 9
Player 2, Sally's Card is: Jack of Clubs
Value is: 11
A point for Player 2, Sally

Turn #: 19
Player 1, Bobby's Card is: Two of Hearts
Value is: 2
Player 2, Sally's Card is: King of Diamonds
Value is: 13
A point for Player 2, Sally

Turn #: 20
Player 1, Bobby's Card is: Three of Spades
Value is: 3
Player 2, Sally's Card is: Queen of Hearts
Value is: 12
A point for Player 2, Sally

Turn #: 21
Player 1, Bobby's Card is: Ace of Spades
Value is: 14
Player 2, Sally's Card is: Eight of Diamonds
Value is: 8
A point to Player 1, Bobby

Turn #: 22
Player 1, Bobby's Card is: Eight of Hearts
Value is: 8
Player 2, Sally's Card is: King of Clubs
Value is: 13
A point for Player 2, Sally

Turn #: 11
Player 1, Bobby's Card is: Three of Clubs
Value is: 3
Player 2, Sally's Card is: Seven of Spades
Value is: 7
A point for Player 2, Sally

Turn #: 12
Player 1, Bobby's Card is: Five of Hearts
Value is: 5
Player 2, Sally's Card is: Four of Diamonds
Value is: 4
A point to Player 1, Bobby

Turn #: 13
Player 1, Bobby's Card is: Four of Spades
Value is: 4
Player 2, Sally's Card is: Jack of Spades
Value is: 11
A point for Player 2, Sally

Turn #: 14
Player 1, Bobby's Card is: Queen of Hearts
Value is: 12
Player 2, Sally's Card is: Three of Spades
Value is: 3
A point to Player 1, Bobby

Turn #: 15
Player 1, Bobby's Card is: Ace of Diamonds
Value is: 14
Player 2, Sally's Card is: Seven of Diamonds
Value is: 7
A point to Player 1, Bobby

Turn #: 16
Player 1, Bobby's Card is: Queen of Spades
Value is: 12
Player 2, Sally's Card is: Six of Clubs
Value is: 6
A point to Player 1, Bobby

Turn #: 17
Player 1, Bobby's Card is: Ten of Diamonds
Value is: 10
Player 2, Sally's Card is: Two of Hearts
Value is: 2
A point to Player 1, Bobby

Turn #: 18
Player 1, Bobby's Card is: Nine of Spades
Value is: 9
Player 2, Sally's Card is: Three of Hearts
Value is: 3
A point to Player 1, Bobby

Turn #: 19
Player 1, Bobby's Card is: Two of Diamonds
Value is: 2
Player 2, Sally's Card is: Ten of Clubs
Value is: 10
A point for Player 2, Sally

Turn #: 20
Player 1, Bobby's Card is: Two of Clubs
Value is: 2
Player 2, Sally's Card is: Eight of Spades
Value is: 8
A point for Player 2, Sally

Turn #: 21
Player 1, Bobby's Card is: Five of Clubs
Value is: 5
Player 2, Sally's Card is: Seven of Hearts
Value is: 7
A point for Player 2, Sally

Turn #: 22
Player 1, Bobby's Card is: King of Spades
Value is: 13
Player 2, Sally's Card is: Ten of Spades
Value is: 10
A point to Player 1, Bobby

Turn #: 23

Player 1, Bobby's Card is: Seven of Diamonds
Value is: 7

Player 2, Sally's Card is: Ten of Spades
Value is: 10

A point for Player 2, Sally

Turn #: 24

Player 1, Bobby's Card is: Five of Diamonds
Value is: 5

Player 2, Sally's Card is: Eight of Spades
Value is: 8

A point for Player 2, Sally

Turn #: 25

Player 1, Bobby's Card is: Jack of Hearts
Value is: 11

Player 2, Sally's Card is: Ace of Hearts
Value is: 14

A point for Player 2, Sally

Turn #: 26

Player 1, Bobby's Card is: Four of Clubs
Value is: 4

Player 2, Sally's Card is: Seven of Spades
Value is: 7

A point for Player 2, Sally

Final Scores:

Player 1, Bobby's score: 14
Player 2, Sally's score: 12

The Winner of WAR is:
Player 1, Bobby

End: Game of WAR!

End of Week 6 Coding Assignment

Turn #: 23

Player 1, Bobby's Card is: Nine of Spades
Value is: 9

Player 2, Sally's Card is: Six of Hearts
Value is: 6

A point to Player 1, Bobby

Turn #: 24

Player 1, Bobby's Card is: Six of Clubs
Value is: 6

Player 2, Sally's Card is: Five of Spades
Value is: 5

A point to Player 1, Bobby

Turn #: 25

Player 1, Bobby's Card is: Four of Clubs
Value is: 4

Player 2, Sally's Card is: Two of Spades
Value is: 2

A point to Player 1, Bobby

Turn #: 26

Player 1, Bobby's Card is: Seven of Spades
Value is: 7

Player 2, Sally's Card is: Eight of Spades
Value is: 8

A point for Player 2, Sally

Final Scores:

Player 1, Bobby's score: 13
Player 2, Sally's score: 13

The Winner of WAR is:
It's a DRAW!

End: Game of WAR!

End of Week 6 Coding Assignment

Turn #: 23

Player 1, Bobby's Card is: King of Diamonds
Value is: 13

Player 2, Sally's Card is: Six of Diamonds
Value is: 6

A point to Player 1, Bobby

Turn #: 24

Player 1, Bobby's Card is: Three of Diamonds
Value is: 3

Player 2, Sally's Card is: Four of Clubs
Value is: 4

A point for Player 2, Sally

Turn #: 25

Player 1, Bobby's Card is: Queen of Diamonds
Value is: 12

Player 2, Sally's Card is: Queen of Clubs
Value is: 12

The values are equal, no score!

Turn #: 26

Player 1, Bobby's Card is: Ten of Hearts
Value is: 10

Player 2, Sally's Card is: Nine of Hearts
Value is: 9

A point to Player 1, Bobby

Final Scores:

Player 1, Bobby's score: 11
Player 2, Sally's score: 13

The Winner of WAR is:
Player 2, Sally

End: Game of WAR!

End of Week 6 Coding Assignment

URL to GitHub Repository: <https://github.com/sw-dev-lisa-s-nh/IntroToJava-week6.git>