

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:

```

1 Card.java X Deck.java X App.java X Player.java
2 package week6final;
3 public class Card {
4
5     //Fields
6     private int value;
7     private String name;
8
9     //Describe card
10    public void describe() {
11        if (value == 11) {
12            System.out.println("Jack of " + name);
13        }
14        else if (value == 12) {
15            System.out.println("Queen of " + name);
16        }
17        else if (value == 13) {
18            System.out.println("King of " + name);
19        }
20        else if (value == 14) {
21            System.out.println("Ace of " + name);
22        }
23        else {
24            System.out.println(value + " of " + name);
25        }
26    }
27    //Getters and Setters
28    public int getValue() {
29        return value;
30    }
31    public void setValue(int value) {
32        this.value = value;
33    }
34    public String getName() {
35        return name;
36    }
37    public void setName(String name) {
38        this.name = name;
39    }
40 }

```

```

1 Card.java X Deck.java X App.java X Player.java
2 package week6final;
3 import java.util.ArrayList;
4 import java.util.Collections;
5 import java.util.List;
6
7 public class Deck {
8     //Fields
9     List<Card> cards = new ArrayList<Card>();
10
11    //Creating the deck
12    List<Integer> value = new ArrayList<Integer>();
13
14    public Deck() {
15        for (int i = 2; i <= 14; i++) {
16            value.add(i);
17        }
18    }
19    String[] name = new String[4];
20
21    name[0] = "Spades";
22    name[1] = "Clubs";
23    name[2] = "Hearts";
24    name[3] = "Diamonds";
25
26    for (int i = 0; i <= 3; i++) {
27        for (int v = 2; v <= 14; v++) {
28            Card card = new Card();
29            card.setName(name[i]);
30            card.setValue(v);
31            cards.add(card);
32        }
33    }
34
35    //Shuffle: randomizes order of cards
36    public void shuffle() {
37        Collections.shuffle(cards);
38    }
39    //Draw: removes and returns top card of the deck
40    public Card draw() {
41        return cards.remove(0);
42    }
43 }

```

```

1 Card.java X Deck.java X App.java X Player.java
2 package week6final;
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class Player {
7     //Fields
8     private List<Card> hand = new ArrayList<Card>();
9     private int score;
10     private String name;
11
12    //Getters and Setters
13    public List<Card> getHand() {
14        return hand;
15    }
16    public void setHand(List<Card> hand) {
17        this.hand = hand;
18    }
19    public int getScore() {
20        return score;
21    }
22    public void setScore(int score) {
23        this.score = score;
24    }
25    public String getName() {
26        return name;
27    }
28    public void setName(String name) {
29        this.name = name;
30    }
31
32    //Describe player
33    public void describe() {
34        System.out.println(name + " plays!");
35    }
36    //Add 1 to score for winning card
37    public void incrementScore() {
38        this.score++;
39    }
40    //Remove and return top card
41    public Card flip() {
42        return hand.remove(0);
43    }
44    //Take deck as argument, return drawn card
45    public void draw(deck) {
46        hand.add(deck.draw());
47    }
48 }

```

```

1 Card.java X Deck.java X App.java X Player.java
2
3 //App.java
4 package week6final;
5
6 public class App {
7     public static void main(String[] args) {
8         Deck deck = new Deck();
9         deck.shuffle();
10
11         Player1 player1 = new Player1("Player1");
12         Player2 player2 = new Player2("Player2");
13
14         //Deal cards to each player
15         for (int i = 0; i < 5; i++) {
16             Card card = deck.draw();
17             player1.hand.add(card);
18             player2.hand.add(card);
19         }
20
21         //Print initial scores
22         player1.describe();
23         player2.describe();
24
25         //Simulate the game
26         for (int i = 0; i < 10; i++) {
27             Card card1 = player1.flip();
28             Card card2 = player2.flip();
29
30             //Compare values and increment scores
31             if (card1.getValue() > card2.getValue()) {
32                 player1.incrementScore();
33             }
34             else if (card2.getValue() > card1.getValue()) {
35                 player2.incrementScore();
36             }
37             else {
38                 //Draw
39                 continue;
40             }
41         }
42
43         //Print final scores
44         player1.describe();
45         player2.describe();
46
47         //Determine the winner
48         if (player1.getScore() > player2.getScore()) {
49             System.out.println("Player1 won!");
50         }
51         else if (player2.getScore() > player1.getScore()) {
52             System.out.println("Player2 won!");
53         }
54         else {
55             System.out.println("Draw!");
56         }
57     }
58 }

```

Screenshots of Running Application:

```

Console X Problems X Debug Shell
<terminated> App [Java Application] /Users/caseykobosh/p2/pool/plugins/org.eclipse.justj.openj
Player1 scored 18 points. Player2 scored 7 points.
Congratulations Player1!

```

URL to GitHub Repository:

<https://github.com/CKobosh/Week-6-Final-Project.git>