

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:

Deck.java Card.java × Player.java App.java

```
1 package WarGame;
2
3 public class Card {
4     private int _value;
5     private String _name;
6
7     public Card() {
8         _name = "";
9         _value = 0;
10    }
11
12    public Card(int value, String name) {
13        _value = value;
14        _name = name;
15    }
16
17    public String getName() {
18        return _name;
19    }
20    public Integer getValue() {
21        return _value;
22    }
23
24    public void setName(String newName) {
25        _name = newName;
26    }
27
28    public void setValue(int newValue) {
29        _value = newValue;
30    }
31
32
33    public void describe() {
34        System.out.println(_name + " and the value is: " + _value);
35    }
36 }
37
38 |
```

```
*Deck.java X Card.java Player.java App.java
1 package WarGame;
2
3 import java.util.ArrayList;
4 import java.util.Collections;
5
6 public class Deck {
7     private ArrayList<Card> deck;
8
9
10    public Deck() {
11        String suits[] = { "Clubs", "Diamonds", "Spades", "Hearts" };
12        String faces[] = { "2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack", "Queen", "King", "Ace" };
13        deck = new ArrayList<>();
14        for (String suit : suits) {
15            for (int i = 0; i < faces.length; i++) {
16                deck.add(new Card(i + 2, (faces[i] + " of " + suits)));
17            }
18        }
19    }
20
21    public Card draw() {
22        return deck.remove(0);
23    }
24
25    public void shuffle() {
26        Collections.shuffle(deck);
27    }
28
29    public void size() {
30        System.out.println(deck.size());
31    }
32 }
33
34
```

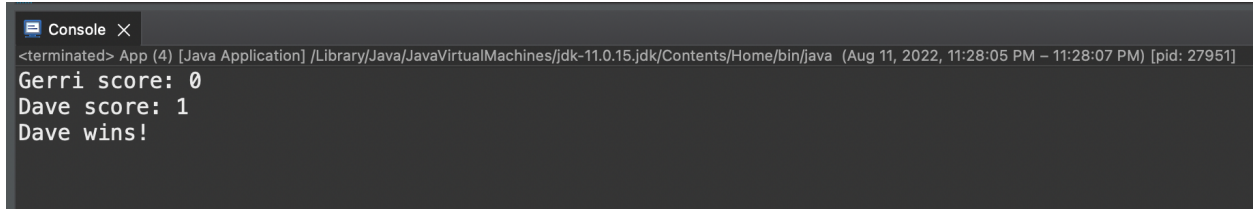
```
*Deck.java  Card.java  Player.java  App.java
1 package WarGame;
2
3 import java.util.ArrayList;
4
5
6 public class Player {
7     private List<Card> hand = new ArrayList<Card>();
8     private int score;
9     private String name;
10
11
12 public Player() {
13     score = 0;
14 }
15
16
17 public void describe() {
18     System.out.println("Player name: " + name + " and my score is: " + score);
19     if (hand.size() > 0) {
20         System.out.println("Cards in Hand: " + hand.size());
21         for (Card card : hand) {
22             card.describe();
23         }
24     }
25 }
26
27 public void draw(Deck mainDeck) {
28     Card card = mainDeck.draw();
29     hand.add(card);
30 }
31
32 public void incrementScore() {
33     score++;
34 }
35
36 public Card flip() {
37     return hand.remove(0);
38 }
39
40 public void setName(String name) {
41     this.name = name;
42 }
43
44 public String getName() {
45     return name;
46 }
47
48 public int getScore() {
49     return score;
50 }
51 }
```

```

1 package WarGame;
2
3 public class App {
4
5     public static void main(String[] args) {
6
7         Deck mainDeck = new Deck();
8
9         Player player1 = new Player();
10        player1.setName("Gerri");
11
12        Player player2 = new Player();
13        player2.setName("Dave");
14
15        mainDeck.shuffle();
16
17        //deal each player 26 cards
18
19        for (int i = 0; i < 52; i++) {
20            if (i % 2 == 0) {
21                player1.draw(mainDeck);
22            } else {
23                player2.draw(mainDeck);
24            }
25        }
26
27        //Iterate threw the cards where each player flips a card.
28        //If their value is higher than the other they get 1 point
29
30        for (int i = 0; i < 26; i++) {
31            Card player1Card = player1.flip();
32            Card player2Card = player2.flip();
33            if (player1Card.getValue() > player2Card.getValue()) {
34                player1.incrementScore();
35            } else if (player2Card.getValue() > player1Card.getValue()) {
36                player2.incrementScore();
37            }
38        }
39
40        // display players score
41
42        System.out.println(player1.getName() + " score: " + player1.getScore());
43        System.out.println(player2.getName() + " score: " + player2.getScore());
44
45        //The winner is...
46
47        if (player1.getScore() > player2.getScore()) {
48            System.out.println(player1.getName() + " wins!");
49        } else if (player2.getScore() > player1.getScore()) {
50            System.out.println(player2.getName() + " wins!");
51        } else if (player1.getScore() == player2.getScore()) {
52            System.out.println("Draw");
53        }
54    }
55 }

```

Screenshots of Running Application:

A screenshot of a Java application console window. The window has a title bar that says "Console" with a close button. Below the title bar, there is a status bar that reads "<terminated> App (4) [Java Application] /Library/Java/JavaVirtualMachines/jdk-11.0.15.jdk/Contents/Home/bin/java (Aug 11, 2022, 11:28:05 PM - 11:28:07 PM) [pid: 27951]". The main area of the console displays the following text: "Gerri score: 0", "Dave score: 1", and "Dave wins!".

```
<terminated> App (4) [Java Application] /Library/Java/JavaVirtualMachines/jdk-11.0.15.jdk/Contents/Home/bin/java (Aug 11, 2022, 11:28:05 PM - 11:28:07 PM) [pid: 27951]
Gerri score: 0
Dave score: 1
Dave wins!
```

URL to GitHub Repository:

<https://github.com/geraldinedepaul17/BackEndClass/tree/main/WarGame>