



PROMINEO TECH

Intro to Java Week 6 Coding Assignment

URL to GitHub Repository: https://github.com/Ete wolde2023/Week-06-Debugging_and_Unit_Tests

URL to Public Link of your Video: https://youtu.be/1_VBIYTe-Eg

Instructions:

1. Follow the **Coding Steps** below to complete this assignment.

- 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.
- Create a new repository on GitHub for this week's assignment and push your completed code to this dedicated repo.
- Create a video showcasing your work:
 - In this video: record and present your project verbally while showing the results of the working project.
 - Easy way to Create a video: Start a meeting in Zoom, share your screen, open Eclipse with the code and your Console window, start recording & record yourself describing and running the program showing the results.
 - Your video should be a maximum of 5 minutes.
 - Upload your video with a public link.
 - Easy way to Create a Public Video Link: Upload your video recording to YouTube with a public link.

2. In addition, please include the following in your Coding Assignment Document:

- The URL for this week's GitHub repository.
- The URL of the public link of your video.

3. Save the Coding Assignment Document as a .pdf and do the following:

- Push the .pdf to the GitHub repo for this week.
 - Upload the .pdf to the LMS in your Coding Assignment Submission.
-



PROMINEO TECH

Intro to Java Week 6 Coding Assignment



Intro to Java Week 6 Coding Assignment

Coding Steps — Java Final Project:

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)



Intro to Java Week 6 Coding Assignment

2. Create a class called App with a main method.
 - a) Instantiate a Deck and two Players, call the shuffle method on the deck.
 - b) Using a traditional for loop, iterate 52 times calling the Draw method on the other player each iteration using the Deck you instantiated.
 - c) Using a traditional for loop, iterate 26 times and call the flip method for each player.
 - d) 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.
 - e) After the loop, compare the final score from each player.
 - f) 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.

3. Tips: Printing out information throughout the game adds value including easier debugging as you progress and a better user experience.
 - a) Using the Card describe() method when each card is flipped illustrates the game play.
 - b) Printing the winner of each turn adds interest.
 - c) Printing the updated score after each turn shows game progression.
 - d) At the end of the game: print the final score of each player and the winner's name or "Draw" if the result is a tie.



PROMINEO TECH

Intro to Java Week 6 Coding Assignment

```
<terminated> Application (2) [Java Application] C:\Users\jermih\p2\pool\plugins\org.eclipse.justi.openjdk hotspot\jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\javaw.exe (Jun. 17, 2023, 12:15:53 p.m. - 12:15:54 p.m.)
Two of Clubs--2
Three of Clubs--3
Four of Clubs--4
Five of Clubs--5
Six of Clubs--6
Seven of Clubs--7
Eight of Clubs--8
Nine of Clubs--9
Ten of Clubs--10
Jack of Clubs--11
Queen of Clubs--12
King of Clubs--13
Ace of Clubs--14
Two of Diamonds--2
Three of Diamonds--3
Four of Diamonds--4
Five of Diamonds--5
Six of Diamonds--6
Seven of Diamonds--7
Eight of Diamonds--8
Nine of Diamonds--9
Ten of Diamonds--10
Jack of Diamonds--11
Queen of Diamonds--12
King of Diamonds--13
Ace of Diamonds--14
Two of Hearts--2
Three of Hearts--3
Four of Hearts--4
Five of Hearts--5
Six of Hearts--6
Seven of Hearts--7
Eight of Hearts--8
Nine of Hearts--9
Ten of Hearts--10
Jack of Hearts--11
Queen of Hearts--12
King of Hearts--13
Ace of Hearts--14
Two of Spades--2
Three of Spades--3
Four of Spades--4
```

When a new Deck was instantiated, the Cards field was populated with the standard 52 cards.



PROMINEO TECH

Intro to Java Week 6 Coding Assignment

```
<terminated> Application (2) [Java Application] C:\Users\ermih\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (Jun. 17, 2023, 12:55:20 p.m. - 12:55:20 p.m.)
Six of Hearts--6
Six of Hearts--6
Four of Diamonds--4
King of Diamonds--13
Four of Spades--4
Queen of Hearts--12
Five of Hearts--5
Two of Diamonds--2
Six of Clubs--6
Nine of Spades--9
Queen of Clubs--12
Jack of Diamonds--11
Ten of Clubs--10
Five of Clubs--5
Two of Spades--2
Eight of Spades--8
Seven of Diamonds--7
Jack of Spades--11
Nine of Hearts--9
Six of Diamonds--6
Jack of Clubs--11
Queen of Diamonds--12
Eight of Clubs--8
Ace of Spades--14
Four of Clubs--4
King of Clubs--13
King of Spades--13
Nine of Clubs--9
Nine of Diamonds--9
Ace of Hearts--14
Eight of Diamonds--8
Seven of Hearts--7
Seven of Spades--7
Ten of Spades--10
Ace of Diamonds--14
Eight of Hearts--8
Two of Hearts--2
Ten of Diamonds--10
Ten of Hearts--10
Four of Hearts--4
Ace of Clubs--14
Queen of Spades--12
Three of Hearts--3
```

The order of the Cards was randomized after the shuffle method was used in the Deck class.



PROMINEO TECH

Intro to Java Week 6 Coding Assignment

```
1 package week06;
2
3 public class Card {
4     int value;
5     String name;
6     String suit;
7     Card(String name, String suit, int value) {
8         this.value = value;
9         this.name = name;
10        this.suit = suit;
11    }
12    public int getValue() {
13        return value;
14    }
15    public void setValue(int value) {
16        this.value = value;
17    }
18    public String getName() {
19        return name;
20    }
21    public void setName(String name) {
22        this.name = name;
23    }
24    public String getSuit() {
25        return suit;
26    }
27    public void setSuit(String suit) {
28        this.suit = suit;
29    }
30    public void describe() {
31        System.out.println(this.name + " of " + this.suit + "--" + this.value);
32    }
33 }
34
```

Class Card with its fields and methods



PROMINEO TECH

Intro to Java Week 6 Coding Assignment

```
5
6 public class Deck {
7     List<Card> cards = new ArrayList<Card>();
8     Deck() {
9         String[] names = {"Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine", "Ten", "Jack", "Queen", "King", "Ace"};
10        String[] suits = {"Clubs", "Diamonds", "Hearts", "Spades"};
11
12        for(String suit : suits) {
13            int count = 2;
14            for(String name : names) {
15                Card card = new Card(name, suit, count);
16                this.cards.add(card);
17                count++;
18            }
19        }
20    }
21
22
23
24
25
26
27 public List<Card> getCards() {
28     return cards;
29 }
30 public void setCards(List<Card> cards) {
31     this.cards = cards;
32 }
33 public void describe() {
34     for(Card card : this.cards) {card.describe();}
35 }
36 public void shuffle() {
37     Collections.shuffle(this.cards);
38 }
39 public Card draw() {
40     Card card = this.cards.remove(0);
41     return card;
42 }
43 }
44
45
46
47
```

Class Deck with its fields and methods



PROMINEO TECH

Intro to Java Week 6 Coding Assignment

```
1 package week06;
2
3 public class Application {
4
5     public static void main(String[] args) {
6         Deck deck = new Deck();
7         deck.describe();
8         deck.shuffle();
9         deck.describe();
10        Card drawnCard = deck.draw();
11        drawnCard.describe();
12    }
13
14 }
15
16
17
18
19
20
```

A separate Application Class was used to test the methods used in the classes, Card and Deck.



PROMINEO TECH

Intro to Java Week 6 Coding Assignment

```
1 package week06;
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     private int score;
9     private String name;
10
11     Player(String n) {
12         score = 0;
13         name = n;
14     }
15
16     public void describe() {
17         System.out.println("player:" + name);
18         for(Card card : hand) {card.describe();}
19     }
20
21     public Card flip() {
22         return hand.remove(0);
23     }
24
25     public void draw(Deck d) {
26         hand.add(d.draw());
27     }
28
29     public void incrementScore() {
30         score += 1;
31     }
32
33 }
34
```

Class Player with its fields and methods



PROMINEO TECH

Intro to Java Week 6 Coding Assignment

```
1 package week06;
2 public class App {
3
4     public static void main(String[] args) {
5         Deck deck = new Deck();
6         Player player1 = new Player("player1");
7         Player player2 = new Player("player2");
8
9         deck.shuffle();
10        for(int i = 0; i <= 26; i++) {
11
12            player1.draw(deck);
13            player2.draw(deck);
14        }
15
16        for(int i = 0; i <= 26; i++) {
17            if(player1.flip().getValue() > player2.flip().getValue()) {
18                player1.incrementScore();
19            } else {
20                player2.incrementScore();
21            }
22        }
23    }
24 }
25
26
27 }
```

In this App Class, two Players and a Deck were instantiated. The shuffle method and a traditional loop were also used.