



Intro to Java Week 6 Coding Assignment

Points possible: 100

URL to GitHub Repository:

URL to Public Link of your Video:

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.

<https://github.com/tstokes433/CardGameOfWar>

- The URL of the public link of your video.

<https://www.youtube.com/watch?v=iH-3DCrqDJ0>

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.



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)



Intro to Java Week 6 Coding Assignment

4. `incrementScore` (adds 1 to the Player's score field)
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.

Week 6 & Week 11 Final Coding Project Rubric

Completion: The purpose of this section is to ensure that the project is implemented completely, and includes all requirements of the Coding Assignment. The addition of **comments in the code** highlights the



PROMINEO TECH

Intro to Java Week 6 Coding Assignment

coder's understanding of the functionality in the code. If the code is not fully functional, **please add comments to outline what has been attempted.**

GitHub: The purpose of this section is to establish a presence on GitHub. Not only does this allow graders to run and verify the project, it provides a platform for potential employers to review. **Please include all code (e.g. Java and/or SQL), any ERD files, and all Coding Assignment documents.**

Errors: The purpose of this section is to evaluate how well this project runs. If there are any errors (compilation or runtime), it is helpful to document what has been attempted to solve the issues. **Please add comments to the code and the Coding Assignment document to explain.**

Video Explanation: The purpose of this section is to demonstrate that the project runs and to demonstrate understanding of the project by verbal description. **Please create a video showcasing this project.** One easy way to do this is to start a Zoom session, share the screen and record a demonstration of the project running, and a verbal explanation of the required functionality.

Completion	All requirements have been attempted. <i>45 points</i>	Did not attempt some (1-2) requirements. <i>25 points</i>	Did not attempt most (3+) requirements. <i>0 points</i>
GitHub	URL provided, and project was able to be cloned and run. <i>35 points</i>	URL provided, but repo is missing the full file structure. OR If SQL/ERD files are missing for MySQL. <i>20 points</i>	URL not provide, URL points to non-existent GitHub repository, OR URL points to an empty GitHub repository. <i>0 points</i>
Errors	No errors. <i>10 points</i>	Minor Errors: code runs, but not all functionality works. Runtime errors exist. <i>5 points</i>	Major Errors: code is not implemented correctly. Compilation errors exist. <i>0 points</i>
Video Explanation	Video showcases all requirements. <i>10 points</i>	Video partially showcases requirements, but not all requirements are included. <i>5 points</i>	Not included. <i>0 points</i>