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

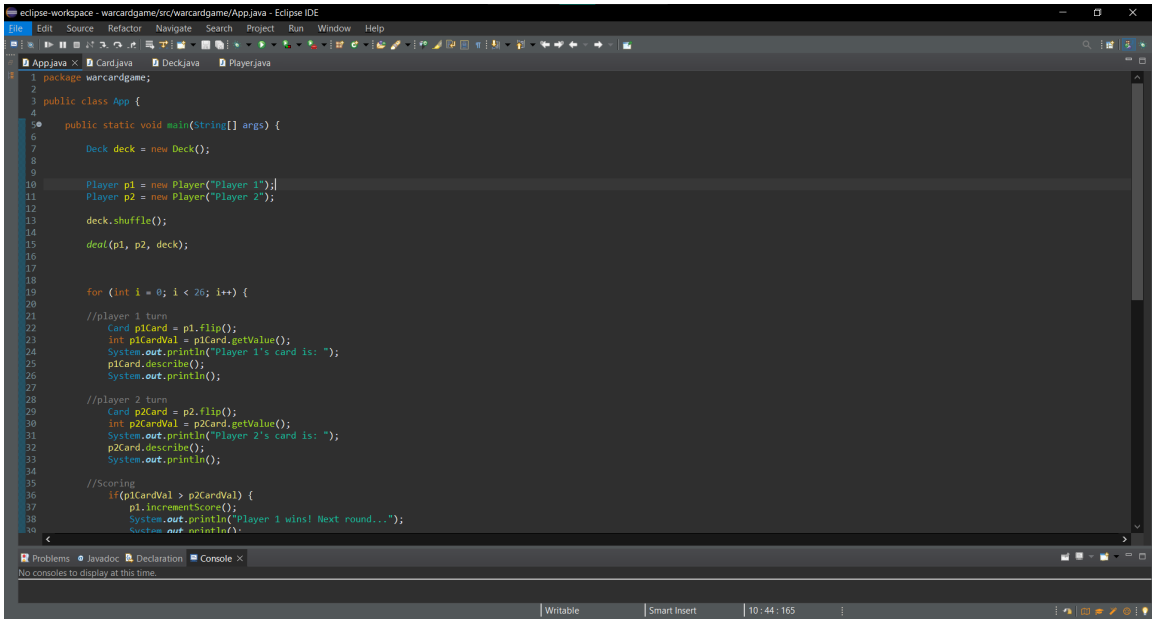
1. describe (prints out information about the player and calls the describe method for each card in the Hand List)
2. flip (removes and returns the top card of the Hand)
3. draw (takes a Deck as an argument and calls the draw method on the deck, adding the returned Card to the hand field)
4. incrementScore (adds 1 to the Player's score field)

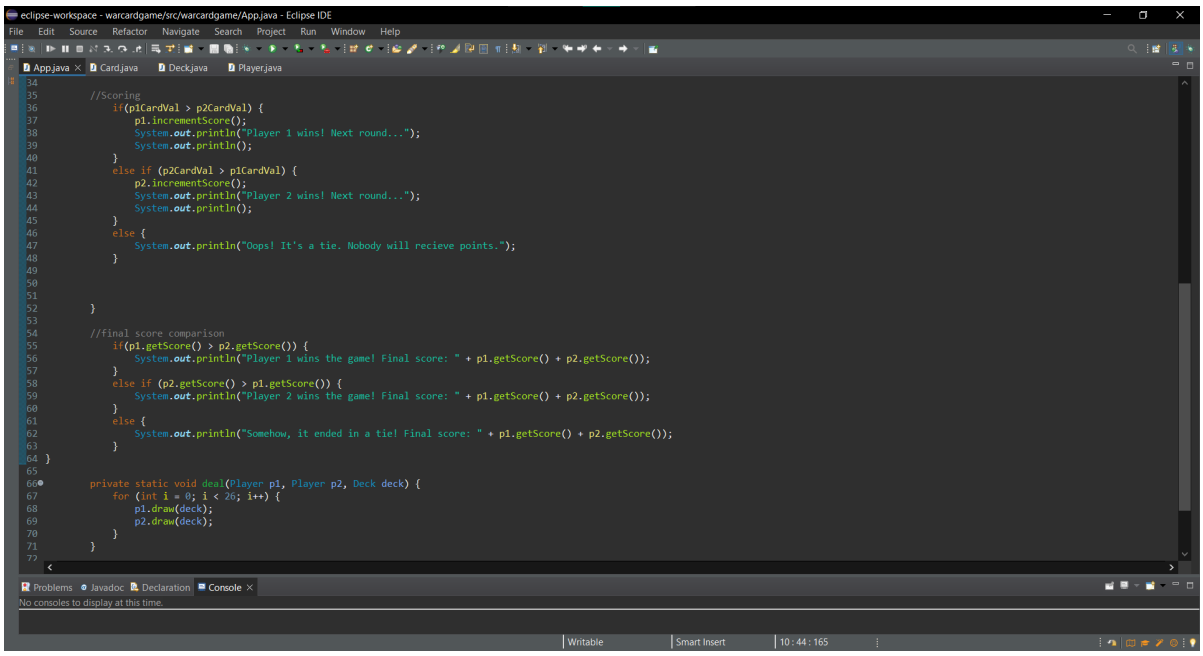
2. Create a class called App with a main method.
3. Instantiate a Deck and two Players, call the shuffle method on the deck.
4. Using a traditional for loop, iterate 52 times calling the Draw method on the other player each iteration using the Deck you instantiated.
5. Using a traditional for loop, iterate 26 times and call the flip method for each player.

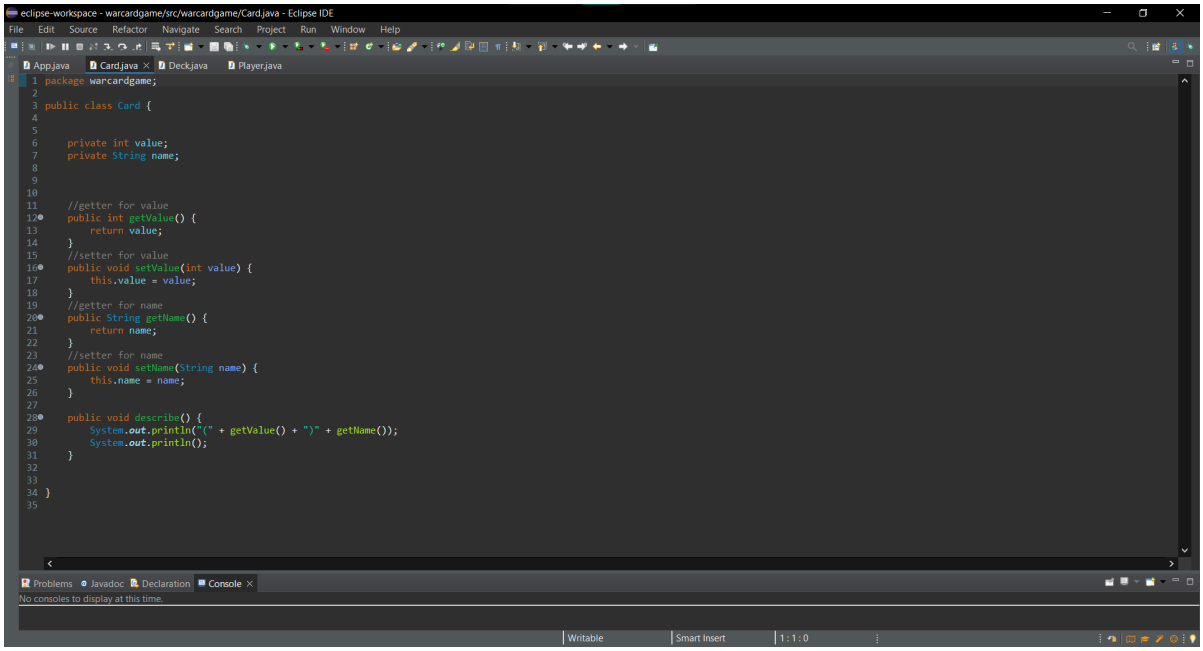
a. Compare the value of each card returned by the two player's flip methods. Call the incrementScore method on the player whose card has the higher value.

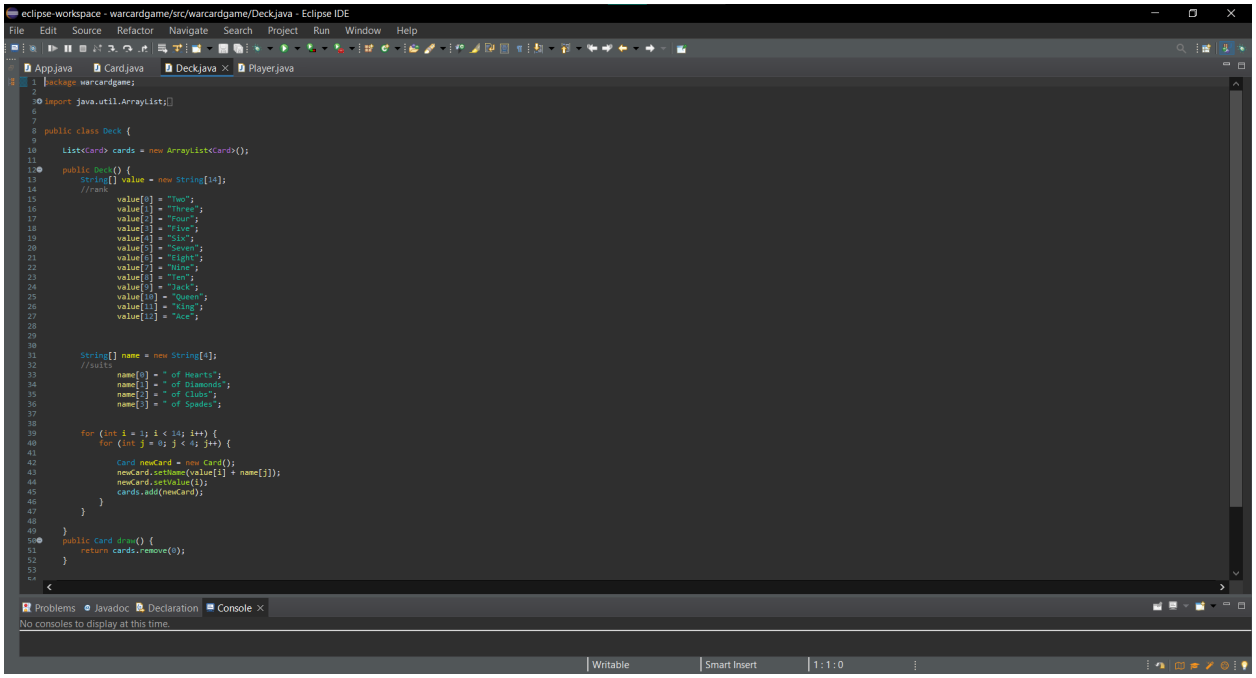
6. After the loop, compare the final score from each player.
7. Print the final score of each player and either "Player 1", "Player 2", or "Draw" depending on which score is higher or if they are both the same.

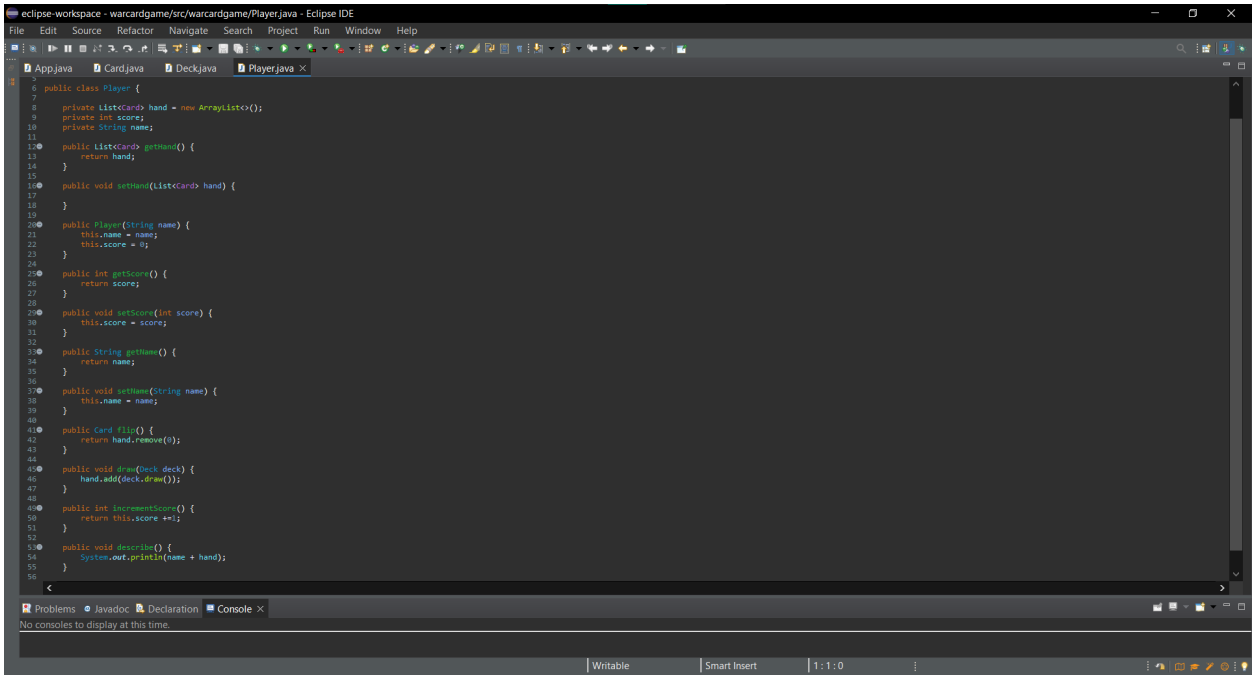
Screenshots of Code:



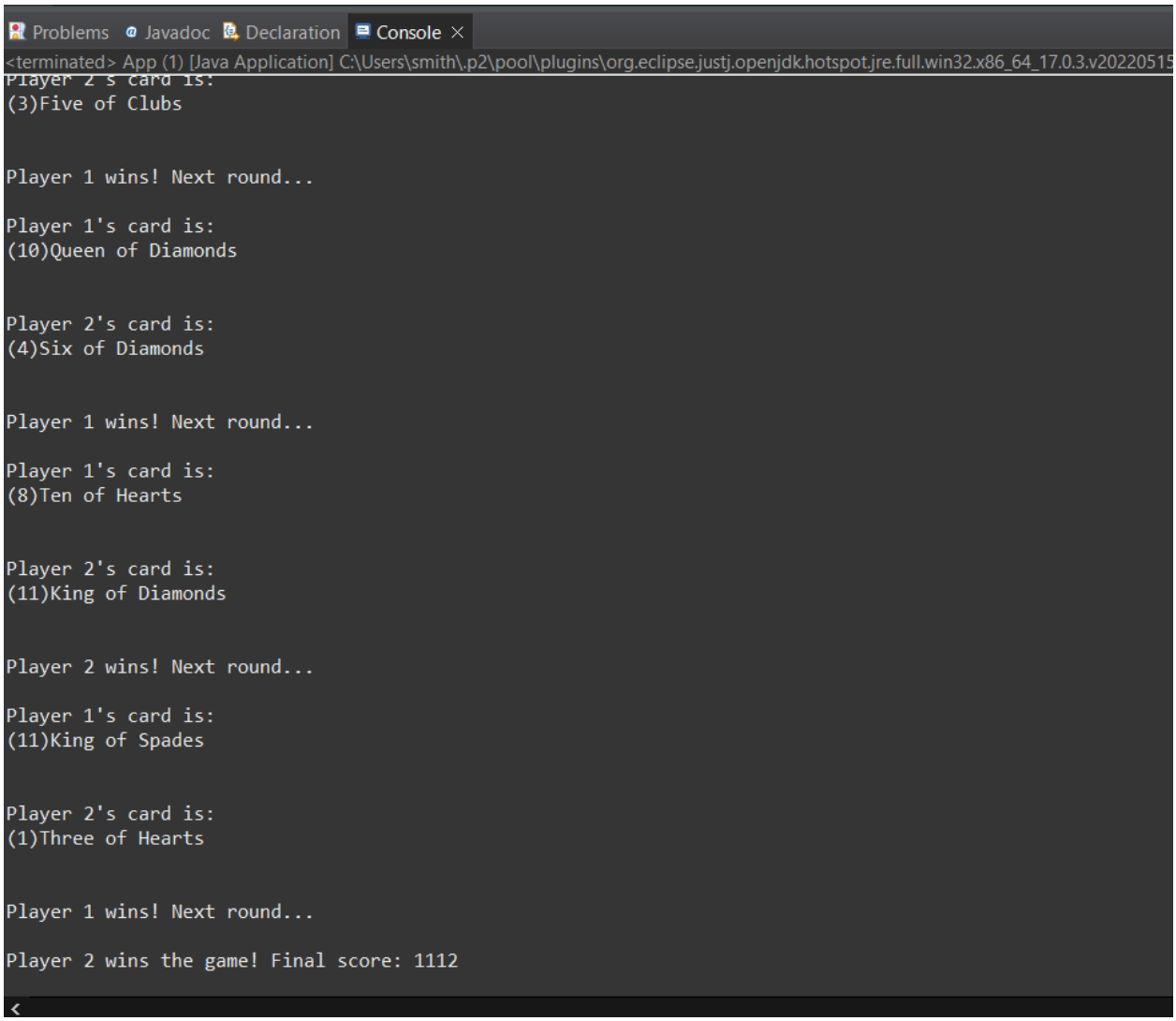

eclipse-workspace - warcardgame/src/warcardgame/App.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
App.java Card.java Deck.java Player.java
//Scoring
if(p1CardVal > p2CardVal) {
p1.incrementScore();
System.out.println("Player 1 wins! Next round...");
System.out.println();
}
else if (p2CardVal > p1CardVal) {
p2.incrementScore();
System.out.println("Player 2 wins! Next round...");
System.out.println();
}
else {
System.out.println("Oops! It's a tie. Nobody will recieve points.");
}
}
//final score comparison
if(p1.getScore() > p2.getScore()) {
System.out.println("Player 1 wins the game! Final score: " + p1.getScore() + p2.getScore());
}
else if (p2.getScore() > p1.getScore()) {
System.out.println("Player 2 wins the game! Final score: " + p1.getScore() + p2.getScore());
}
else {
System.out.println("Somehow, it ended in a tie! Final score: " + p1.getScore() + p2.getScore());
}
}
private static void deal(Player p1, Player p2, Deck deck) {
for (int i = 0; i < 26; i++) {
p1.draw(deck);
p2.draw(deck);
}
}
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable Smart Insert 10 : 44 : 165


eclipse-workspace - warcardgame/src/warcardgame/Card.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
App.java Card.java Deck.java Player.java
package warcardgame;
public class Card {
private int value;
private String name;
//getter for value
public int getValue() {
return value;
}
//setter for value
public void setValue(int value) {
this.value = value;
}
//getter for name
public String getName() {
return name;
}
//setter for name
public void setName(String name) {
this.name = name;
}
public void describe() {
System.out.println("(" + getValue() + ")" + getName());
System.out.println();
}
}
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable Smart Insert 1 : 1 : 0


eclipse-workspace - warcardgame/src/warcardgame/Deck.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
App.java Card.java Deck.java Player.java
package warcardgame;
import java.util.ArrayList;
public class Deck {
List<Card> cards = new ArrayList<Card>();
public Deck() {
String[] value = new String[14];
//rank
value[0] = "Two";
value[1] = "Three";
value[2] = "Four";
value[3] = "Five";
value[4] = "Six";
value[5] = "Seven";
value[6] = "Eight";
value[7] = "Nine";
value[8] = "Ten";
value[9] = "Jack";
value[10] = "Queen";
value[11] = "king";
value[12] = "Ace";
String[] name = new String[4];
//suits
name[0] = " of Hearts";
name[1] = " of Diamonds";
name[2] = " of Clubs";
name[3] = " of Spades";
for (int i = 1; i < 14; i++) {
for (int j = 0; j < 4; j++) {
Card newCard = new Card();
newCard.setName(value[i] + name[j]);
newCard.setValue(i);
cards.add(newCard);
}
}
}
public Card draw() {
return cards.remove(0);
}
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable Smart Insert 1 : 1 : 0


eclipse-workspace - warcardgame/src/warcardgame/Player.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
App.java Card.java Deck.java Player.java
public class Player {
private List<Card> hand = new ArrayList<>();
private int score;
private String name;
public List<Card> getHand() {
return hand;
}
public void setHand(List<Card> hand) {
}
public Player(String name) {
this.name = name;
this.score = 0;
}
public int getScore() {
return score;
}
public void setScore(int score) {
this.score = score;
}
public String getName() {
return name;
}
public void setName(String name) {
this.name = name;
}
public Card flip() {
return hand.remove(0);
}
public void draw(Deck deck) {
hand.add(deck.draw());
}
public int incrementScore() {
return this.score +=1;
}
public void describe() {
System.out.println(name + hand);
}
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable Smart Insert 1 : 1 : 0

Screenshots of Running Application:



URL to GitHub Repository:

[GitHub repository Link]