

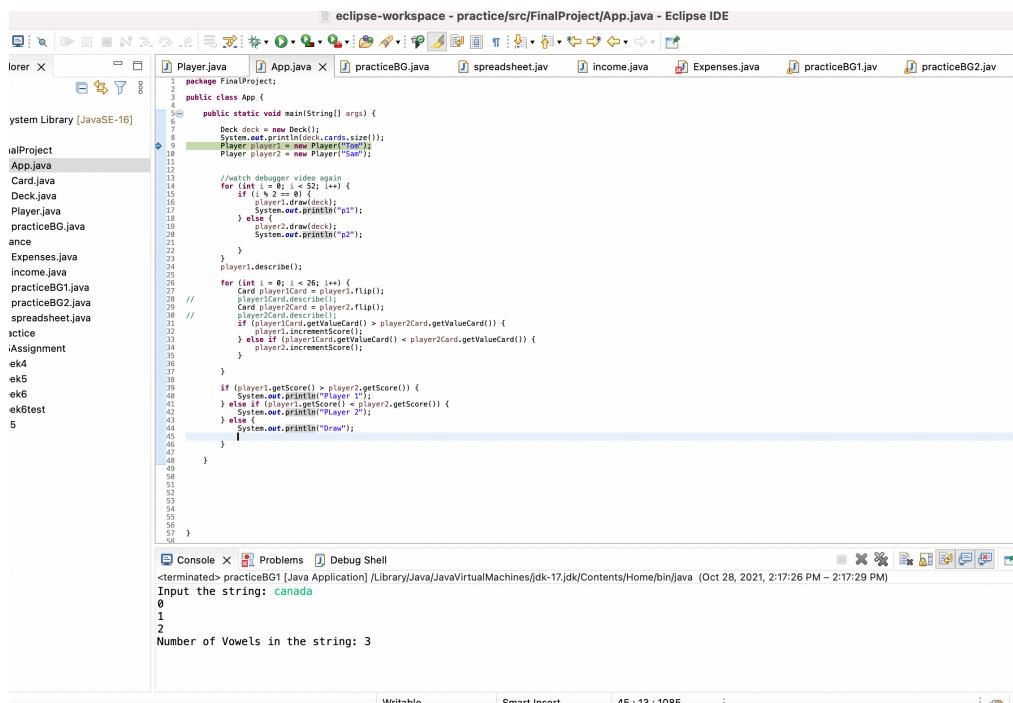
Coding Steps:

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 - practice/src/FinalProject/App.java - Eclipse IDE

Player X
  Card.java
  Deck.java
  Player.java
  practiceBG0.java
  Expenses.java
  income.java
  practiceBG1.java
  practiceBG2.java
  spreadsheet.java
  action
  Assignment
  e4
  e5
  e6
  e6test
  5

public class App {
    public static void main(String[] args) {
        Deck deck = new Deck();
        System.out.println("Deck size: " + deck.cards.size());
        Player player1 = new Player("Sam");
        Player player2 = new Player("Sam");

        //watch debugger video again
        for (int i = 0; i < 52; i++) {
            if (i < 26) {
                player1.draw(deck);
                System.out.println("p1");
            } else {
                player2.draw(deck);
                System.out.println("p2");
            }
            player1.describe();
        }
        for (int i = 0; i < 26; i++) {
            Card player1Card = player1.flip();
            player1Card.describe();
            Card player2Card = player2.flip();
            player2Card.describe();
            if (player1Card.getValueCard() > player2Card.getValueCard()) {
                player1.incrementScore();
            } else if (player1Card.getValueCard() < player2Card.getValueCard()) {
                player2.incrementScore();
            }
        }

        if (player1.getScore() > player2.getScore()) {
            System.out.println("Player 1");
        } else if (player1.getScore() < player2.getScore()) {
            System.out.println("Player 2");
        } else {
            System.out.println("Draw");
        }
    }
}

Console X Problems Debug Shell
<terminated> practiceBG1 [Java Application] /Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home/bin/java (Oct 28, 2021, 2:17:26 PM – 2:17:29 PM)
Input the string: canada
0
1
2
Number of Vowels in the string: 3

```

App.java

```

1 package FinalProject;
2
3 public class Card {
4     private String nameCard;
5     private int valueCard;
6     public Card() {}
7     public Card(String name, int value) {
8         this.nameCard = name;
9         this.valueCard = value;
10    }
11    public void describe() {
12        System.out.println(nameCard + ":" + valueCard);
13    }
14    public int getValueCard() {
15        return valueCard;
16    }
17    public void setValueCard(int valueCard) {
18        this.valueCard = valueCard;
19    }
20    public String getNameCard() {
21        return nameCard;
22    }
23    public void setNameCard(String nameCard) {
24        this.nameCard = nameCard;
25    }
26}
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43

```

Card.java

```

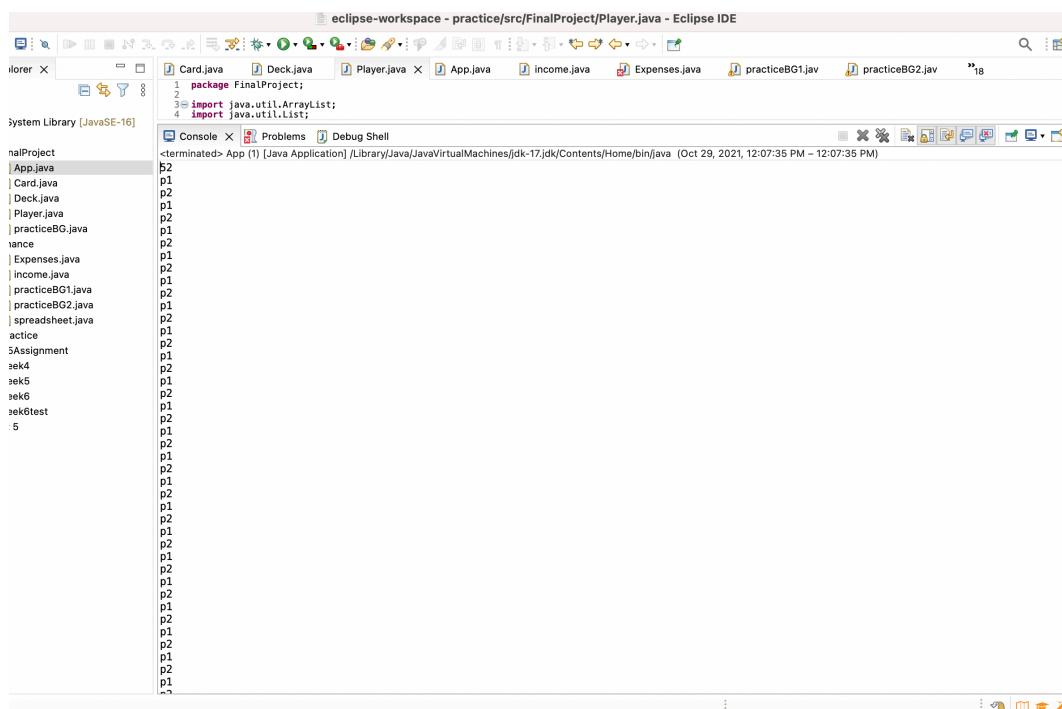
1 package FinalProject;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class Player {
7     private ArrayList hand;
8     public Player() {
9         hand = new ArrayList();
10    }
11    public void addCard(Card card) {
12        hand.add(card);
13    }
14    public void removeCard(Card card) {
15        hand.remove(card);
16    }
17    public void printHand() {
18        for (Card card : hand) {
19            System.out.println(card.describe());
20        }
21    }
22
23    public static void main(String[] args) {
24        Player p1 = new Player();
25        Player p2 = new Player();
26
27        p1.addCard(new Card("Two of Clubs", 0));
28        p1.addCard(new Card("Four of Clubs", 2));
29        p1.addCard(new Card("Six of Clubs", 4));
30        p1.addCard(new Card("Eight of Clubs", 6));
31        p1.addCard(new Card("Ten of Clubs", 8));
32        p1.addCard(new Card("Queen of Clubs", 10));
33        p1.addCard(new Card("Ace of Clubs", 12));
34
35        p1.printHand();
36
37        p2.addCard(new Card("Three of Hearts", 1));
38        p2.addCard(new Card("Five of Hearts", 3));
39        p2.addCard(new Card("Seven of Hearts", 5));
40        p2.addCard(new Card("Nine of Hearts", 7));
41        p2.addCard(new Card("Jack of Hearts", 9));
42        p2.addCard(new Card("King of Hearts", 11));
43
44        p2.printHand();
45
46        System.out.println("Player 1");
47        System.out.println("Player 2");
48    }
49}

```

Caption

Deck.java

Screenshots of Running Application:



Caption

The screenshot shows the Eclipse IDE interface with the title bar "eclipse-workspace - practice/src/FinalProject/Player.java - Eclipse IDE". The left sidebar shows a project tree with packages like FinalProject, App.java, Card.java, Deck.java, Player.java, practiceBG.java, finance, Expenses.java, income.java, practiceBG1.java, practiceBG2.java, spreadsheet.java, practice, w5Assignment, week4, week5, week6, week6test, and init 5. The right pane displays the Java code for Player.java:

```
1 package FinalProject;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class Player {
7     String name;
8     int score;
9     List<Card> hand = new ArrayList<Card>();
10    public Player(String name) {
11        this.name = name;
12        this.score = 0;
13    }
14    public void describe() {
15        System.out.println(name + ": " + score);
16        for (int i = 0; i < hand.size(); i++) {
17            hand.get(i).describe();
18        }
19    }
20    public Card flip() {
21        Card card = hand.get(0);
22        hand.remove(0);
23        return card;
24    }
25    public void draw(Deck deck) {
26        this.hand.add(deck.draw());
27    }
28    public void incrementScore() {
29        score++;
30    }
31    public int getScore() {
32        return score;
33    }
34    public void printHand() {
35        for (Card card : hand) {
36            card.describe();
37        }
38    }
39}
40
```

The code defines a Player class with methods for describing its state, flipping a card from its hand, drawing a card into its hand, incrementing its score, and printing its hand. The code uses Java's List and ArrayList classes.

Player.java

URL to GitHub Repository: