Course:	INFO 3134, W2025
Professor:	Janice Manning and Madhavi Mohan
Assignment:	Project # 1: Inheritance and Polymorphism
Due Date:	Friday, March 7, 2025 at 11:59 pm

CardGame Application

In this project, you will design a class hierarchy to model a card game application. A player can play the Game of 21 many times, until he chooses not to play or until the deck is empty. The player is playing against the computer. In the Game of 21, the player gets some choice of moves, but the computer's moves are based on current standings in its hand when compared to the player's hand. Here are the rules for the Game of 21:

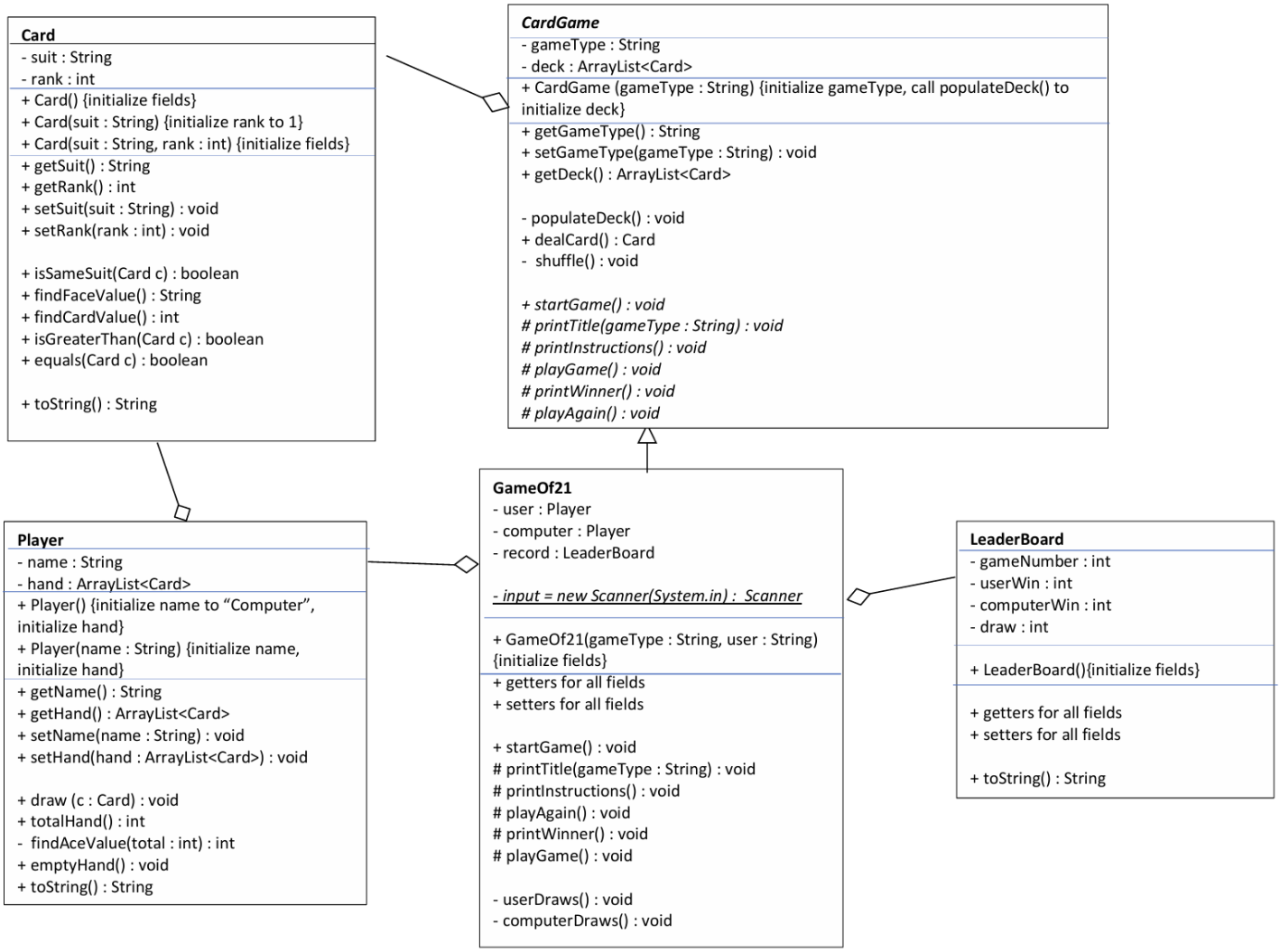
Game of 21
https://en.wikipedia.org/wiki/Blackjack The objective of the game is getting the closest to 21 without going over. Each player is dealt with 2 cards. The user can then decide whether to draw another card. The computer does the same. Face cards have a value of 10 and Aces can be high or low (that is, 11 points or 1 point) depending on achieving the closest score to 21 without going over. After 2 or more cards have been drawn by each player, the score is totaled, and a winner is declared.

The CardGame application includes the following classes:

- Card, an object class for a playing card
- Player, an object class for a player of a card game, and initializes an ArrayList of Card objects, which represents a Player's hand
- CardGame, an abstract superclass which holds behaviours that are common to its subclasses, and initializes an ArrayList of Card objects, which represents a deck of 52 cards
- GameOf21, a subclass of CardGame, which includes Player objects and methods to play the Game of 21 against the computer
- LeaderBoard, an object class to hold game data which is printed at the end of all game play
- CardGameTester, a tester class which allows a user to play the Game of 21 against the computer

See the UML diagram for the classes on the next page. Then, carefully read the program specifications that follow before you begin coding.

UML for CardGame Application:



Program Specifications:

1. **Card class is an object class for a Card. Card is an aggregate class for Player and CardGame**
   - The isSameSuit() method compares the suits of this card to the parameter and returns true if the suits are the same.
   - The findFaceValue() method checks the rank of this card, and returns one of the following face values:
     - {"Ace", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine", "Ten", "Jack", "Queen", "King"}
   - The findCardValue() method checks the rank of this card, and returns its card value based on the following:
     - {return card values as their ranks for ranks between 1-10; return card values as 10 for ranks 11-13}
   - The isGreaterThan() method compares the rank of this card to the parameter and returns true if this card's rank is greater than the parameter card's rank.
   - The equals() method compares the rank of this card to the parameter and returns true if the ranks are the same.
   - The toString() method returns a String that represents the current state of a Card object, using the card's face value and its suit.

2. **Player class is an object class for a Player. Player is an aggregate class for GameOf21 ~~and the GameOfWar~~.**
   - The draw(Card c) method adds a Card object to a Player's hand, which is dealt from the CardGame class deck.
   - The totalHand() method finds the total value of a Player's hand based on the Cards' values. This method also checks if a Card is an Ace and, if it is, calls the findAceValue() helper method.
   - The findAceValue(int total) is a helper method which checks if the Player's current hand total plus 11 goes over 21, then returns ace as 1, otherwise returns ace as 11.
   - The emptyHand() method empties a Player's hand of cards.
   - The toString() method returns a String representation of a Player's current hand of Card objects.

3. **CardGame class is an abstract superclass**
   - The populateDeck() method loads an ArrayList<Card> deck with 52 Card objects made up of 4 suits, each with 13 ranks. Once the deck is populated, the helper method shuffle() is called.
   - The shuffle() method shuffles the ArrayList<Card> deck in random order.
   - The dealCard() method returns and removes the "top" Card object from the deck, that is, the Card at Index 0.
   - The following abstract methods will be implemented in CardGame subclasses and their implementation will be specific to the card game:
     - The startGame() method will begin a Card Game by calling the printTitle(), printInstructions(), and playGame() methods.
     - The printInstructions() method prints the instructions of a Card Game.
     - The playGame() method draws Players' cards, and calls printWinner() and playAgain() methods.
     - The printWinner() checks the points of each player and prints the winner or if the game is a draw.
     - The playAgain() method checks that the deck has cards; then prompts the user to play again, gets the user's response to play again and, if yes, empties the Players' hands, ~~populates the deck~~, updates the LeaderBoard game number, and plays the game again; otherwise prints the LeaderBoard object and an exit message.

4. **GameOf21 class is a subclass of CardGame**
   GameOf21 implements the abstract methods of CardGame as described above, with the following additions:
   - The playGame() method deals two cards to each player by having each Player object get cards from the deck by calling its respective draw() method (userDraws() and computerDraws(). Print each player, showing their two cards. Then, call userDraws() and computerDraws() repeatedly while the user's hand is less than 21 or the user stands. Then call printWinner() and playAgain() methods.
   - The printWinner() method checks the total points of each winner. The player with the points closest to 21 without going over is the winner. A draw occurs if the points are equal. Update the corresponding attribute of the LeaderBoard object in each case: draw, user wins, computer wins.
   - The userDraws() method asks the user if he wants to draw another card and, if he does, draws another card; otherwise the user stands. The method returns a char, holding the user's choice, to the method.
   - The computerDraws() method will draw another card if the user's total hand is >= to the computer's hand; otherwise the computer stands. See the additional sample output to determine the cases for when the computer stands!

5. **LeaderBoard class is an aggregate class for GameOf21**
   - The toString() method returns a String representation of a LeaderBoard object which summarizes all games played.

To Test the CardGame Application use CardGameTester.java

1. Print a message to the screen and get the user's name.
2. Create a CardGame object.
3. Begin the card game, by calling startGame().

CardGame Application - Marking Scheme

Description	Available	Awarded
Overall Coding Style - Each class includes: o A complete document header o Javadoc style documentation for all methods o Adequate commenting within methods o Correct use of access modifiers for fields, methods, and constructors, as specified on the UML o Correct use of indenting and white space throughout classes		**/4**
Card class • Fields are declared with appropriate types and modifiers • Three constructors are present and each correctly initializes Card fields • Getters and setters are present for all fields • Method header correct and method body correctly implemented for all methods: o isSameSuit(), findFaceValue(), findCardValue(), isGreaterThan(), equals(), toString()		**/7**
Player class • Fields are declared with appropriate types and modifiers • Two constructors are present and each correctly initializes Player fields • Getters and setters are present for all fields • Method header correct and method body correctly implemented for all methods: o draw(), totalHand(), findAceValue(), emptyHand(), toString()		**/8**
CardGame abstract class • Fields are declared with appropriate types and modifiers • One constructor is present and each correctly initializes CardGame fields • Getters and setters are present, as specified • Method header correct and method body correctly implemented for the following methods: o populateDeck(), dealCard(), shuffle() • Abstract method headers are present and correctly declared: o startGame(), printInstructions(), playGame, printWinner(), playAgain()		**/7**
GameOf21 class, a subclass of CardGame • Fields are declared with appropriate types and modifiers • One constructor is present and each correctly initializes CardGame and GameOf21 fields • Getters and setters are present for all fields • Abstract methods are correctly implemented for the GameOf21: o startGame(), printInstructions(), playGame, printWinner(), playAgain() • Method header correct and method body correctly implemented for the helper methods: o userDraws(), computerDraws()		**/10**
LeaderBoard class • Fields are declared with appropriate types and modifiers • One constructor is present and each correctly initializes LeaderBoard fields • Getters and setters are present for all fields • Method header correct and method body correctly implemented for: o toString()		**/5**
CardGameTester class • Prints a message to the screen, and get the user's name • creates a CardGame object • Begins the card game, by calling startGame()		**/4**
CardGame Application Overall • Project runs successfully (no compiler errors, no compiler warnings, no exceptions)		**/10**
Total Marks		**/55**

See the sample outputs for the Game of 21 below.

Sample Output #1

```
Welcome to the CardGame Application!
-------------------------------------
Enter your name: Stewart
-------------------------------------
How to play the Game of 21
-------------------------------------
 . . . game instructions go here . . .

Stewart has 2 cards:
1) Nine of Spades
2) Two of Hearts
11

Computer has 2 cards:
1) Queen of Diamonds
2) Jack of Spades
20

Do you want another card (Y/N)? y
Stewart has 3 cards:
1) Nine of Spades
2) Two of Hearts
3) Seven of Hearts
18

Computer has 3 cards:
1) Queen of Diamonds
2) Jack of Spades
3) Ace of Spades
21

Do you want another card (Y/N)? n
Stewart stands.

--------------------------------
Game of 21 Game #1
--------------------------------
Stewart points:       18
Computer points:      21
--------------------------------
Computer wins!
--------------------------------

Play another game (Y/N)?
```

Sample Output #2

```
Welcome to the CardGame Application!
-------------------------------------
Enter your name: Stewart
-------------------------------------
How to play the Game of 21
-------------------------------------
 . . . game instructions go here . . .

Stewart has 2 cards:
1) Six of Diamonds
2) Nine of Hearts
15

Computer has 2 cards:
1) Two of Hearts
2) Eight of Diamonds
10

Do you want another card (Y/N)? y
Stewart has 3 cards:
1) Six of Diamonds
2) Nine of Hearts
3) Nine of Spades
24

Computer stands.

--------------------------------
Game of 21 Game #1
--------------------------------
Stewart points:       24
Computer points:      10
--------------------------------
Computer wins!
--------------------------------

Play another game (Y/N)? n

--------------------------------
Leader Board Summary
--------------------------------
Games Played:          1
User Wins:             0
Computer Wins:         1
Draws:                 0
--------------------------------

Thanks for playing! Bye!
```

Sample Output #3

```
Welcome to the CardGame Application!
---------------------------------------
Enter your name: Stewart
---------------------------------------
How to play the Game of 21
---------------------------------------
 . . . game instructions go here . . .

Stewart has 2 cards:
1) King of Clubs
2) Ten of Hearts
20

Computer has 2 cards:
1) Queen of Hearts
2) Five of Clubs
15

Do you want another card (Y/N)? n
Stewart stands.

--------------------------------
Game of 21 Game #1
--------------------------------
Stewart points:       20
Computer points:      15
--------------------------------
Stewart wins!
--------------------------------

Play another game (Y/N)? y
Stewart has 2 cards:
1) Nine of Spades
2) Nine of Diamonds
18

Computer has 2 cards:
1) Queen of Diamonds
2) Three of Clubs
13

Do you want another card (Y/N)? y
Stewart has 3 cards:
1) Nine of Spades
2) Nine of Diamonds
3) Three of Hearts
21

Computer stands.

--------------------------------
Game of 21 Game #3
--------------------------------
Stewart points:       21
Computer points:      13
--------------------------------
Stewart wins!
--------------------------------

Play another game (Y/N)?
```

Sample Output #4

```
Welcome to the CardGame Application!
---------------------------------------
Enter your name: Stewart
---------------------------------------
How to play the Game of 21
---------------------------------------
 . . . game instructions go here . . .

Stewart has 2 cards:
1) Ten of Hearts
2) Queen of Hearts
20

Computer has 2 cards:
1) King of Hearts
2) King of Diamonds
20

Do you want another card (Y/N)? n
Stewart stands.

--------------------------------
Game of 21 Game #1
--------------------------------
Stewart points:       20
Computer points:      20
--------------------------------
Game is a draw.
--------------------------------

Play another game (Y/N)? y
Stewart has 2 cards:
1) Jack of Clubs
2) Ace of Diamonds
21

Computer has 2 cards:
1) Two of Spades
2) King of Spades
12

--------------------------------
Game of 21 Game #2
--------------------------------
Stewart points:       21
Computer points:      12
--------------------------------
Stewart wins!
--------------------------------

Play another game (Y/N)?
```