Single Player Blackjack

Choice of Tooling

- My blackjack implementation is in Java because Java is an object-oriented programming language. Blackjack naturally has discrete objects such as cards, decks, and players, so designing those objects in Java as classes is useful.
- I used Eclipse to write and run my code. Eclipse is popular for Java programs and has an easy interface for running and debugging programs.

Design Choices

- Card
    - I store the name, number, and value of the card so when a player draws that card I can easily print out its name to the console and append its value to the player's current sum.
    - I considered checking if an invalid number is passed into the constructor and initially decided that the constructor should throw an exception if the number is less than 1 or greater than 13, but then realized that I call the constructor anyways and since I know the rules of the game I don't need to check those conditions.
    - I used global variables for parameters such as the number of copies of each number, minimum value, and maximum value in case someone does not play with the standard deck and wants to change those values.
- ShuffledDeck
    - I allow the user to specify the number of decks they want to use in case they want more of each card or want to play with more players. This was intended for longer games. Given the recommended time constraint, I decided not to implement multiplayer blackjack.
    - I shuffle the cards using an algorithm that ensures that every permutation of the cards is equally likely to appear, thus making shuffling completely random. I decided to implement the shuffling algorithm myself with the underlying data structure as an array so I could efficiently swap elements.
    - I use a linked list to represent the cards in the deck because a linked list removes the 0th index element in O(1) time.
- Player and PlayBlackjack
    - I keep a list of scores rather than just one score for the player. This is in case the player draws an ace, in which case the player has multiple options between the scores they have.

    - For every instance of when a user needs to provide input, I decompose that instance so that I can re-prompt the user if the user enters invalid input.
    - When prompting the user for how many decks they want to play with, I allow them to press Enter to play with a default deck of 1. This is what most users probably want, but I give users who want to change it the opportunity to do so.

How to Play

- Download all of the files, including the .java files and .class files, onto your computer.
- In Terminal, go to the directory where you have downloaded those files.
- Run **java PlayBlackjack**
    - The Java class files are already created for your convenience. However, if you would like to refresh those files, run **javac PlayBlackjack.java**
- Enjoy!