

Card games are an excellent example of when a particular data structure is suited to a task. Stacks emulate the behavior of a person drawing from the top of a deck of cards, so we're going to use that as the basis for a simplified version of the game Crazy Eights.

For minimal passing credit: Create a Deck class of Cards that extends ArrayStack. You can use the Card and ArrayStack classes provided with this assignment (note that this ArrayStack extends ArrayList—you can use functions in ArrayList to make your life easier). The Deck should have (at least) two methods: one that fills the ArrayStack with a standard set of 52 playing cards, and one that shuffles the deck (use Java's Collections.shuffle function for this). Write a driver class that instantiates a Deck, fills it, prints the cards in the stack, shuffles the Deck, then prints the cards again.

For C credit: Create the Deck class described in the D section. In addition, write a driver class called CDriver.java that splits the deck in into two ArrayLists. Then write a function that takes in an ArrayList and a single card (you can pick the card), and determines which cards in the ArrayList “match” it, and returns an Array of matching cards. The easiest way to compare two cards is to implement Comparable and compareTo, or modify the provided .equals() method. Keep in mind that for Crazy Eights, a “match” is either the same suit, the same value, **or** if the card is an 8. An 8 can match anything. Run this function over each half of the deck, and in main() print the matched cards from each hand.

For B credit: Do everything described in the C section. In addition, create a Player class that has an ArrayStack (or ArrayList) for cards in the player's hand. The Player should incorporate the function from the C level: compare a single card to the cards in the players hand and return an array of matches.

Write a driver that gives two players 5 cards each, puts the remaining cards in a Draw stack, then takes the top card from the Draw stack and checks to see how many matches each player has. Determine the “winner” based on which player has more matches to the comparison card. Print each player's matches to the screen.

For A credit: Implement everything in the B version of the code, but modify the driver so that it plays through the entire game of Crazy Eights (i.e., keep going until one Player has no more cards). When it's a Player's turn, the player should check their hand to see if they have cards that match the top of the discard stack. If they do, they should pick one of the cards that match and play it (this should remove the card from the Player's hand and return it to main so it will be added to the Discard pile). If not, they should draw cards until they have a card that they can play. If the Draw pile is empty, the player with the fewest cards left wins.

For Extra Credit: Instead of ending the game when the Draw pile is empty, take the Discard pile, shuffle it, and return it to the Draw pile, using the top card as the start of the Discard pile. The game will only end when one player runs out of cards.

What I need:

1. A narrative about what your code does (which version of the code you implemented, how found pairs, etc), and how you tested it.
2. A zipped file of the **commented** code. As always, if it doesn't compile, it fails (would be better to turn in “Hello World” in that case).

Crazy Eights

(instructions modified from <http://www.bicyclecards.com/article/top-three-classic-family-card-games/>)

Players: 2

Objective: Be the first player to get rid of all your cards.

Rules:

Deal 5 cards to each player.

Place the rest of the cards face down in the center of the table in the Draw pile, then turn the first card up and place it as the starter for the Discard pile.

Starting with Player 1, each player places one card face up on the starter pile. Each card played (other than an eight) must match the card showing at the top of the starter pile in suit or or value. So if the King of Clubs is the starter, the card played must either be another King, or be any card from the Clubs suit. If unable to play a card, the player draws cards from the top of the Draw pile until a play is possible.

If the Draw pile runs out, shuffle the cards underneath, turn them face down, and make them the new Draw pile. Take the top card from the draw pile to start the Discard pile

All eights are wild. An eight may be played at any time in turn, and if an eight is on the discard pile, any card may be matched to it.

Winning:

The first player without cards (or the player with the fewest cards when the Draw deck runs out, depending on which version of the code you are implementing) wins the game!