Mid-Project

Due March 25th, 11:59 PM

100 points

CS 2235

Data Structures and Algorithms

1. Create a program to simulate the card game War. (If you don't know the rules to War, a video link will be posted on Moodle to explain them.) You may utilize the War, Card, Deck, and Player classes developed in class. Make sure your deck uses all 54 cards in a standard deck of cards. Use appropriate data structures. Use appropriate object-oriented programming. Finally, make use of methods (i.e. don't put everything in the main).Note: Do not prompt the user for ANY input. Assignments submitted with prompts for user input will not be graded.
    a. First, create the game for two players. Print the cards as they are played during the game (including wars) on separate lines. Also, print the winner of the round on a new line. Identify which player won (i.e. Player x has won this round, Player x has won the war, etc.).
    b. Print the number of cards each player has left at each step after the round winner has been declared.
    c. Keep track of the number of rounds in the game and display this information at the end, along with which player won the game. The game must end only when 1 player has all the cards.
    d. Include screenshots of a normal turn during the game, a basic war, a tied war and the summary of the game.

2. In a separate file, play 1,000 games of two-player war. (It is recommended that you comment out your round by round print statements.)Use Java to find:
    a. The average number of rounds per game.

b. The average number of wars per game.
c. The number of wins by each player.
d. Attach a screenshot of the summary output.
e. In a separate Word file, comment about the results. Based on what you know about the war game, were the results what you expected? Theorize why or why not.

Extra Credit (worth up 20 points):

Expand your War game to include 3 players. Your code must be able to conclude the game with only a single winner and deal with scenarios where multiple players draw the same card again in the war. You **must** attach a screenshot of a 3 player war code being implemented and clearly identify this section of code in your submission to receive credit. Submit this as a separate file.

Scoring

1. 60%- War game runs correctly, all information is printed correctly and screenshots are included.
2. 40%- 1,000 games of War played correctly, averages calculated and displayed correctly.