Programming Project - Stacks & Queues

Part 1 - Implementing a Stack and a Queue - Project6.java

1) Download Project6Starter.zip and copy the files into an Eclipse project. Note that you should **not** add Project6.java to any package and the LInkedStack and LinkedQueue classes **must** remain in the jsjf package.
2) Complete the implementation of the LinkedStack & LinkedQueue classes in the jsjf package.
3) Your code should work with the provided sample test driver. You should make sure your code is correct by performing additional tests - the test driver may not cover all the cases that will be tested when you submit your code.

Part 2 - Team Beat The Dealer Card Game - BeatDealer.Java

1) Write a class called StackHand which implements the HandOfCards interface using the LinkedStack class from the jsjf package. Include a method called "play" which removes a card from the hand and returns the card.
2) Copy your Deck & PlayingCard classes from the previous project. Your deck class should generate a deck of 52 playing cards (with no duplicates).
3) Implement a simple game where there are a set number of players playing as a team versus the dealer.
   - Deal several **StackHands** of cards - one for each player and one for the dealer.
   - In each turn, the dealer shows their card then each player shows their card.
   - Combine the value of each player card. The player's win the hand if their total is greater than the dealer's card's value times the number of players. Otherwise the dealer wins.
   - Ties always go to the dealer
4) Your program **should use a queue** to keep track of which player should be playing their card at any point. In each round, display the cards that were played.
5) Once all of the cards have been played, print the number of hands the won by the dealer and the players and display who won.
6) Your code must make use of the LinkedStack (in StackHand) and LinkedQueue (to keep track of the players) classes you implemented in part 1. Implementing the game without using the required data structures will result in a grade of 0 for that part of the assignment.
7) You may want to write a separate player class to simplify your program, but this is not required.

Submission requirements:

- Implement both parts in the same Java project
- Use the file names listed in each part (Project6.java & BeatDealer.java) for your 2 driver classes. Both driver classes should be in the default package. If they are not named appropriately & included in the default package you will lose 25 points.
- Include your name as a comment at the top of each source code file
- Make good use of whitespace/comments to make your implementation clear.
- Make sure you follow the package requirements in the project description. Failure to follow the directions may result in a grade of 0 or the loss of a significant number of points.
- In a well-formatted .doc, .pdf, or .txt file, briefly describe your implementation (including a class diagram), give sample output for both parts, and include the code you use to play the card game.
- Zip your entire Eclipse project (including .class files). Do not use .rar.
- Include your first and last name in the .zip filename
- Upload your implementation/output document & zipped project **separately** to Canvas
- (Optional) Turn in a hard copy of your implementation document

The implementation document will make up 30% of the grade for this project.

Be prepared to demo your project in class following the due date.