605.201 Mini-Project 2:

Please do the following to complete this assignment.

Purpose:
The purpose of this project is to provide non-trivial practice in the use of Java programming constructs and have a bit of fun doing it.

Resources Needed:

You will need a computer system with JRE, Java SE, and JDK. You may optionally use a Java IDE for example NetBeans, Eclipse, etc.

Application Design:

Try to keep the length of the design between 1 and 2 pages. The following subjects should be discussed in this order:

1. Include a UML class diagram, or other diagram of your choice, or written description of your solution.

2. What alternative design approaches were considered and why were they rejected?

Source file:

Each public class must be contained in a separate Java source file. Only one source file will have a main() method and this source will be named BlackjackGameSimulator.java. Other source/class names are up to you following the guidelines specified so far in the course. The format of the Java source must meet the general Java coding style guidelines discussed so far during the course. Pay special attention to naming guidelines, use of appropriate variable names and types, variable scope (public, private, protected, etc.), indentation, and comments. Classes and methods should be commented with JavaDoc-style comments (see below). Please use course office hours or contact the instructor directly if there are any coding style questions. JavaDocs: Sources should be commented using JavaDoc-style comments for classes and methods. Each class should have a short comment on what it represents and use the @author annotation. Methods should have a short (usually 1 short sentence) description of what the results are of calling it. Parameters and returns should be documented with the @param and @return annotations respectively with a short comment on each. JavaDocs must be generated against every project Java source file. They should be generated with a -private option (to document all protection-level classes) and a –d [dir] option to place the resulting files in a javadocs directory/folder at the same level as your source files. See the JavaDocs demonstration for more details. When you submit the one pdf to Canvas, include a few screenshots of the JavaDoc output.

Collaboration:

It is encouraged to discuss technical or small design parts of this project with your fellow students. However, the resulting design and implementation must be your own. When in doubt, ask during office hours or contact your instructor.

Program Specification:

This project involves writing a program to simulate a blackjack card game. You will use a simple console-based user interface to implement this game.

A simple blackjack card game consists of a player and a dealer. A player is provided with a sum of money with which to play. A player can place a bet between $0 and the amount of money the player has. A player is dealt cards, called a hand. Each card in the hand has a point value. The objective of the game is to get as close to 21 points as possible without exceeding 21 points. A player that goes over is out of the game. The dealer deals cards to itself and a player. The dealer must play by slightly different rules than a player, and the dealer does not place bets. A game proceeds as follows: A player is dealt two cards face up. If the point total is exactly 21 the player wins immediately. If the total is not 21, the dealer is dealt two cards, one face up and one face down. A player then determines whether to ask the dealer for another card (called a "hit") or to "stay" with his/her current hand. A player may ask for several "hits." When a player decides to "stay" the dealer begins to play. If the dealer has 21 it immediately wins the game. Otherwise, the dealer must take "hits" until the total points in its hand is 17 or over, at which point the dealer must "stay." If the dealer goes over 21 while taking "hits" the game is over and the player wins. If the dealer's points total exactly 21, the dealer wins immediately. There is only one deck of cards. It is up to Player / Dealer to decide whether the value of an Ace is 1 or 11, based on the hands they have. When the dealer and player have finished playing their hands, the one with the highest point total is the winner. Play is repeated until the player decides to quit or runs out of money to bet. You must use an object-oriented solution for implementing this game.

Follow the Assignment and Project Submission document