Programming Assignment #5

Due: 11/20/2022 23:59:59

Topic: simulating a simple but complete Poker game

Objective: modifying the implemented classes to use class inheritance

Description:

In this assignment, we will modify the Poker (Soha) game in the third assignment to make use of class inheritance. Your modification will be based on sample C++ programs that are given to you. You have the freedom to base your modification on the sample solution of the third assignment or on the program that you wrote for the third assignment.

The rules and requirements in this assignment remain the same as in the third assignment. However, we will use class inheritance to relate `SHPlayer` and `SHDealer`. We mentioned that it makes more sense to inherit the `SHDealer` class from the `SHPlayer` class since they have a "*Is-A*" relation. The `SHDealer` class needs to know how to start a new game, how to print cards to the screen, how to compute the total for a hand of cards, etc, and these functions are already implemented by the `SHPlayer` class. Thus, a `SHDealer` is a `SHPlayer`.

In this assignment, you are asked to change the containment relation (object within object) to an inheritance relation. The `SHDealer` class will simply inherit and use most data members and member functions in the `SHPlayer` class except for the `start()` function. The `SHDealer` class needs to behave differently since it needs to check the number of cards left that can be dealt out before starting a new game.

Assignment Directory: `/usr/local/class/oop/assign/assign5`

You are given a sample program from the third assignment in
`/usr/local/class/oop/sample/assign3`
to start with.

What to Hand in:

You need to submit a complete and working program electronically. Please make comments in your code to help the TA to understand what you have changed.

Implementation Notes:

This is a one-week assignment but it takes less than half of an hour to make the changes if you know where to change. No matter you choose to modify the provided sample or to modify your own, you are encouraged to take this chance to understand and digest the sample code and compare it with yours.