**CSC123 Programming Project 4**

**Video Poker Game**

**Problem Description**

|  |  |  |
| --- | --- | --- |
| **Class** | **Attributes** | **Actions** |
| **Player** | Bankroll bankroll  PokerGame pg  Bet bet | Initialize the bankroll.  Add coins. Bet and Play.  Discard.  Display a Hand.  Quit. Display final results.  Present a menu. |
| **PokerGame** | Player player  Bet bet  Bankroll bankroll  Hand hand | View initial hand.  Discard or hold cards. Update bankroll |
| **Bet** | int bet | Get the bet.  Set the bet. |
| **Deck** | Card deck[] | Shuffle the deck.  Deal a card. |
| **Card** | int suit  int value | Get the suit.  Get the value, i.e., rank.  Get the name of a card. |
| **Hand** | Card [] hand Deck deck | Evaluate the hand. Deal a new hand.  Update a hand. Give the hand. |
| **Bankroll** | int bankroll | Get the bankroll.  Set the bankroll.  Change the bankroll. |

The poker game application was designed and developed in Chapter 11. The game consists of seven interacting classes: Player, PokerGame, Bet, Deck, Card, Hand, and Bankroll. The details of these classes are summarized as follows.

These classes are implemented in Chapter 11, where the Player class provides a text-based user interface. In Chapter 20, the Player class is reimplemented , replacing text-based input and output with a GUI of buttons, labels, and pictures.

**Requirements**

1. Carefully read Chapter 11 and Chapter 20 to understand the class design and implementation of all seven classes for the video poker game.
2. Implement all classes. You can copy the implementation from the textbook, but you have to debug the program if any errors exist.
3. Add a menu to the GUI-based Player class. Your menu should include the following menu items: Check the coin balance, Add coins to the bankroll, Start a new game, Exit the game.
4. You may add some other features, for example, save the coin balance to a file so that the balance can be loaded when you re-run your program. This is optional.
5. Test the complete implementation.

**Submission**

The completed project should be included in a single file and submitted onto the blackboard. The hardcopy or email submission will not be accepted. Your submission should include:

1. The class diagram.
2. The Java code for all classes, all put in a single word or PDF file.
3. You have to demonstrate your program execution to the instructor. The demonstration time will be arranged during class meeting and/or the instructor office hours.