linked list (pointer)
operator overloading
copy constructor
destructor

EECE 2560: Fundamentals of Engineering Algorithms
Department of Electrical and Computer Engineering

Project Flip

Write a program that allows the user to play the card game *flip*. Flip is played by one player with a standard deck of 52 cards. The game has the following steps:

no joker

1. The cards are shuffled three times. spade, club, diamond, heart; A, 2, 3, 4, …, 10, J, Q, K
2. The player draws 24 cards from the top of the deck, without looking at them, and places them face down on the table.
3. The player can either select a card to turn over, or end the game. If a card is turned over, the player:
   (a) receives 10 points for an ace,
   (b) receives 5 points for a king, queen or jack,
   (c) receives 0 points for an 8, 9 or 10,
   (d) loses half their points for a 7, → floor(points/2)
   (e) loses all their points for a 2, 3, 4, 5 or 6, and
   (f) receives 1 point extra, in addition to the above, for a heart.
4. The goal is to end the game with the most points.

Part a

part b is abt giving memory to the player

Fully implement a `card` class that stores a single card. A card includes a value and a suit (club, diamond, heart or spade). The class should at least include (1) a constructor, (2) `setValue()` and `setSuit()` functions, (3) `getValue()` and `getSuit()` functions, and (4) an overloaded << operator to print a card's value and suit.

Fully implement a `deck` class that stores the cards in a deck in order. A deck of cards should be implemented using a **linked list** of `nodes`, each of which contains a single card. The `deck` object should contain a pointer to the first card in the deck. The class should at least include (1) a constructor that creates a deck with all the cards in order (ace-king, club-diamond-heart-spade), (2) an overloaded << operator that prints the cards in the deck, and (3) a `shuffle()` function that shuffles the cards in the deck (Use any algorithm to shuffle the cards that puts the cards in a random order).

Include a `main` function that initializes a deck and prints all the cards in the deck before shuffle and after shuffle.

deck

```
node<card> * front;   // data member
constructor;          // init a new deck of 52 cards
.
```

♦♥♣♠ + ♡♢♤♧