4: A Deck of Cards

CSCI 212 Fall 2015

1 Goals

- To write a simple C++ program.
- To use an enumerated type and a class type.
- To create a standard deck of 52 cards.

2 Overview

Cards. Many games are played with a deck of cards. Decks come in many varieties, specialized for particular games. In this project, however, we will focus entirely on the standard 52-card bridge deck with four suits (spades, hearts, diamonds, and clubs) and 13 cards in each suit (1 to 10, jack, queen, and king). The 1 is often called an "ace".

This is a 2-phase program. In the first phase (program 4), you will generate a deck of cards, store the cards in an array, and print them. In the second phase (program 5), you will shuffle the card array and print the deck again. The algorithm used gives a fair random shuffling order.

Please, DON'T complicate this. It should be a very short and simple program.

3 An Enum Type and a Class

- Define an enumerated type named **SuitT** that lists a constant meaning "suit unknown", followed by the four suits of a card deck. Use UPPER CASE for constants in your enumeration.
- Global constants are OK, but be sure that you do not use global variables in your code. Copy these global CONSTANT declarations at the top of your file. They define the point values and names for the suits and ranks. (Note: slot 0 of each array is unused.)

```
const int points[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10};
const char* suits[] = {"Unknown","Spades","Hearts","Diamonds","Clubs"};
const char ranks[] = {"-A23456789TJQK"};
```

- Define a `class` named **CardT**that represents a playing card. Your type should have three data members, a suit (represented by an enum constant), a rank (represented by a character from the `ranks` array), and a short int point value.

4 Functions to Implement.

In main() Do the following things in your main program:

- Use banner, bye, and fatal, where appropriate.
- Declare a an output stream and an array of 52 CardT to hold the cards.
- Write nested for loops to generate 52 cards, in order. The outer for loop will cycle through the four suits, the inner for loop will cycle through the 13 ranks for each suit. Each time around the loop, construct one card and store it in the array. When all the cards have been generated, loop through the array again and display the cards to the screen using print(cout).

In your CardT class Define these functions

- `CardT::CardT( ){}`
  This is a do-nothing constructor (a null constructor). You need it to create an array of CardT.
- `CardT::CardT( SuitT suit, int rank )`
  Initialize the data members corresponding to the two parameters and use the rank to compute the value for the points, using the global constant array of point values.
- `void print ( ostream& out )`
  Print the card's data members (suit, rank, and point value) stream `out`. Print this in a readable format such as "Q Hearts: 10". Try to make things line up in neat columns.