Santa Clara University
School of Engineering

## On the Provided Code

The provided code uses some C++ features that you may not be familiar with (yet).

The “enum” class gives better names for integral constants. C++ compiler translates those names into integers whose values can be explicitly specified in the class. This is a better coding practice than using “const int” or “#define”.

A colon “:” after the contructor’s prototype and function body begins the “initializers” for the class. In Poker::Poker, we initialize the const string array that name the Poker hands. We took care to use names in the same order of the enum class members.

The class “string” is provided by all C++ compilers. It’s a better abstraction to the concept of string than the traditional “char *” that’s widely used in C. We highly encourage you to use this “type” to represent the string concept. There are legitimate reasons to revert back to using “char *”, but you should not encounter those in this course.

The key word “static” means “only one copy” in C++. All static members in a class exist independent of its instantiations. This means you may refer to static members without using an object and use only the class name.

## Techniques for Ranking Poker Hands

Poker ranking logic appears intimidating in its complexity. It’s easier to analyze the hand first by counting how many cards are in each suit (Spade, Heart, Diamond, Club) and how many cards are in each “rank” (Ace, King, Queen, Jack, …, 4, 3, 2). Once analyzed this way, the code becomes straight-forward as a cascading if-then-else logic. We have already implemented RankHand in poker.cxx. What you need to implement is some of the helper functions.

### How about Ties?

The lab does not require you to resolve ties. This is just for your background information.

If two hands are of the same category, Poker has established tie-breaking rules. While the rules are similar in concept, their logics are slightly different for each category. They all involve a deeper dive into that array with the count of card face values. For example, to break the tie for two “High Card” hands, you pick the best card (in terms of high face value) and see which hand is better. If that ties again, pick the second-best card, and so on. This can be done trivially by sorting that “face count array” in descending order first. The same works for tie-breaking Flushes and Straights.

Santa Clara University
School of Engineering

Pairs tie-break with the face value of the paired card first. Then the rest as in High Card. Similarly for Two Pairs, Three of the Kind, Four of a Kind.

Full House first compare the triple, then the pair.

We have two general strategies to compare two poker hands. We can rank both by category then run them through the tie-breaking rules. Or we can encode the tie-breaking into the ranking directly. Basically, as we rank a hand, we also compute the tie-breaking value as if it will be tied with another similarly ranked hand. If we combine both into a single value, then we do a simple "less than" or "greater" than operation to decide which hand wins.

Note that Poker considers two hands are tied if the tie-breaker yields the same result. You code should be careful in declaring winner. If hand A is "not less than" hand B, it does not mean A has won. It could be a tie.