# CSI 2372 – Lab Task 10

**Abdorrahim Bahrami**

## Inheritance in C++


Your task in this lab is to get yourself familiarized with inheritance and its related concepts in C++. You need to implement a class for a simplified version of the game poker. To make it simple and organized we show cards with simply using a string of length 2. The first letter is the rank of the card, which can be A, 2, 3, 4, 5, 6, 7, 8, 9, T, J, Q, K. We use T for 10, to be able to represent the cards with two letters. The second letter is the suit, which can be D for diamonds, C for clubs, S for spades, or H for hearts. All letters are capital.

You should have these numbers in the class to manage the game.

| |
|---|
| The number of players. |
| The list of cards in the hand of each player. (You need a list of a list for this) |
| The list of cards on the table. |

You also need to hold the deck of cards.

You should have the following method in your class.

| |
|---|
| Poker (The default constructor) shuffles the deck of cards and initializes the arguments above. |
| The number of players, default is 2. |
| The empty list of cards for each player. |
| The empty list of cards on the table. |

| |
|---|
| Poker (The user constructor) shuffles the deck of cards and initializes the arguments above. |
| The number of players is given as an argument. |
| The empty list of cards for each player. |
| The empty list of cards on the table. |

The method add_card, which receives the index of a player and adds the top card on the deck to the hand of that player.

The method add_to_table, which adds a card from top of the deck to the table.

The following methods receives a list of five cards and returns a Boolean

IsStraightFlush (Returns true if all cards are from the same suit and their ranks is in order)

Example: [AH, 2H, 3H, 4H, 5H] True

Example: [TS, JS, QS, KS, AS] True

Example: [7D, 9D, JD, TD, 8D] True (cards might not be given in the rank order)

Example: [8C, 9C, TC, 2C, 3C] False

IsFourofaKind (Returns true if there are four cards of the same rank)

Example: [9H, 6S, 9D, 9C, 9S] True

Example: [2H, 6S, 5H, 8C, 9S] False

IsFullHouse (Returns true if there are three cards of the same rank and two cards of another rank)

Example: [JH, 7S, 7D, JD, JC] True

Example: [2H, 6D, 5D, 8C, 8S] False

IsFlush (Returns true if all five cards have the same suit)

Example: [JH, 7H, 8H, 2H, 5H] True

Example: [3D, 6D, 3H, 8C, 8S] False

IsStraight (Returns true if the five cards are in the order of their rank, suits are not important)

Example: [AD, 2S, 3D, 4C, 5H] True

Example: [TS, JS, QH, KH, AC] True

Example: [7S, 9D, JD, TH, 8H] True (cards might not be given in the rank order)

Example: [8C, 9D, TC, 2D, 3H] False

IsThreeofaKind (Returns true if there are three cards of the same rank)

Example: [TH, 6S, TD, TC, QS] True

Example: [2H, 6S, KH, 8C, QS] False

IsTwoPairs (Returns true if there are two pairs of cards of the same rank)

Example: [TH, 6S, AS, TC, 6D] True

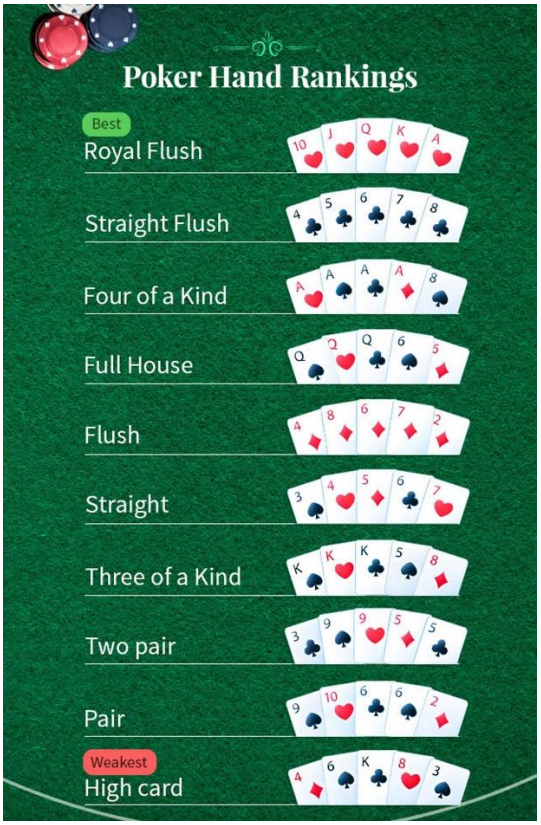
Example: [2H, 6S, 9H, 8C, QS] False

IsOnePair (Returns true if there is a pair of cards of the same rank)

Example: [TH, 6S, AS, KC, KD] True

Example: [2H, 6S, 9H, 8C, QS] False

The following picture shows how hands in Poker are formed. Note that here we mixed straight flush and royal flush in one. If none of the hands above is formed, the hand is called High Card.



Inherits a class from the Poker class above called TexasHoldem, which has the following methods.

TexasHoldem (The default constructor), which initializes the lists like the parent class.

TexasHoldem (The user constructor), which initializes the lists like the parent class.

deal, which deals two cards to the hand of each player and five cards on the table.

hands, which returns a list of what each player has.

For each player, this method should choose five cards from the total seven cards on the table and two cards in the player's hand to form the best hand the player can have.

For example, if the first player has JD and KS in their hand and the second player has JH and TC in their hands and the cards on the table are 2C, JC, KD, 5C, AC, then this method should return ["Two Pairs", "Flush"].

You might find the following piece of code useful, which creates all possible 5 cards out of 7.

```
#include <algorithm>


int i;
vector <string> cards; //place all cards on the table and in the hand of the player into this vector
vector <int> select(7);
select[0] = select[1] = 0;
select[2] = select[3] = select[4] = select[5] = select[6] = 1;
do
{
    vector <string> pick;
    for(i = 0; i < 7; ++i)
        if (select[i] == 1) pick.push_back(cards[i]);

   //pick is one possible selection, you can check it here
}
while(next_permutation(select.begin(), select.end()));
```