**Hands Documentation**

**By Chris Ricchi**

**Introduction**

Welcome to the Hands program, a trial where we simulate 10,000 drawings of a 5-card poker hand and calculate the probability of having the following card combinations:

* One pair
* Three of a kind
* Four of a kind
* Straight
* Flush
* Full House
* Straight Flush
* Royal Flush
* High Card

Our simulation in the Hands program will clearly determine the probability of each hand being drawn, and demonstrate how absurdly rare some of the potential hands can be in a standard poker game.

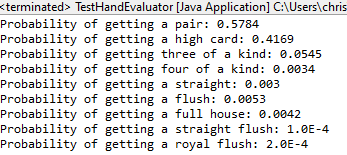
**Using Hands Program**

This program functions with the use of three classes: **Card, HandEvaluator,** and **TestHandEvaluator**. The first class, **Card**, is used to create a Card object that contains a unique face, suit, and value in realistic bounds. The second class, **HandEvaluator**, is used to handle the creation of a hand. This contains methods to initialize a standard 52-card deck, to draw a card, to draw a hand with a customizable amount of cards per hand, and lastly multiple Boolean methods to check the condition(s) of the hand that’s been drawn.

* isPair() – This method checks the value of two cards within a hand to determine if they are a pair or not.
* hasHighCard() – This method checks if the given hand has no combination of a win condition (AKA, the worst hand.)
* hasThreeOfAKind() – This method checks the value of three cards within a hand to determine if there are three cards of a kind or not in the given hand.
* hasFourOfAKind() – This method checks the value of four cards within a hand to determine if there are four cards of a kind or not in the given hand.
* hasStraight() – This method first sorts the hand by value and then checks if the hand contains five cards in sequential rank.
* hasFlush() – This method checks if the given hand contains five cards of all the same suit, but not of sequential rank.
* hasFullHouse() – This method checks if the given hand contains any three cards of the same number or face value, as well as any other two cards of the same number or same face value.
* hasStraightFlush() – This method checks if the given hand contains five cards of sequential rank, and of the same suit.
* hasRoyalFlush() – This method checks if the given hand contains an Ace, King, Queen, Jack, and a 10 all of the same suit.

**Understanding the Results**

By default, the TestHandEvaluator program will simulate 10,000 drawings and calculate the probability of each hand occurring. With such a large amount of simulations, we can expect the calculated probabilities to be relatively accurate in terms of real life conditions.



**In-Depth Results**

Probability of getting a pair: **57.8%**

Probability of getting a high card: **41.7%**

Probability of getting three of a kind: **5.04%**

Probability of getting a four of a kind: **0.34%**

Probability of getting a straight: **0.3%**

Probability of getting a flush: **0.53%**

Probability of getting a full house: **0.42%**

Probability of getting a straight flush: **0.01%**

Probability of getting a royal flush: **0.02%**

In further analysis, these results make sense when you consider what has to happen for each hand to occur. The two worth noting are straight flush and royal flush, which are two of the rarest hands in poker.