List of Inputs, Outputs, and Processing Required

INPUTS

* + None (unless you count the random function grabbing the unix timestamp)

OUTPUTS

* + Deck ToString
  + Shuffled Deck ToString
  + Two hand ToStrings

PROCESSING

* + Creating a deck
  + Shuffling the deck
  + Dealing cards into hands

Identification of Classes and Their Responsibilities

**Class name: Card**

Responsibilities

* + Knowing Face
  + Knowing Suit
  + Formatting into a ToString

**Class name: Deck**

Responsibilities

* + Generating every possible card to make a complete deck
  + Shuffling self
  + Keeping track of cards that have been dealt
  + Formatting into a ToString

**Class name: Hand**

Responsibilities

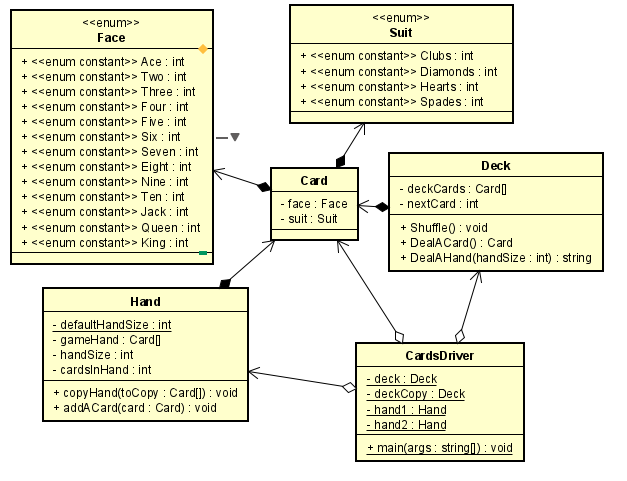
* + Holding cards for a user that can be added and removed
  + Preventing illegal deals
  + Formatting into a ToString

**Class name: CardsDriver**

Responsibilities

* + Creating a deck
  + Displaying the deck
  + Shuffling the deck
  + Copying the deck
  + Displaying the copied deck
  + Dealing 2 hands
  + Displaying the hands

UML Class Diagram



Algorithms

**Class: Deck**

Operation:Shuffle()

START

FOREACH(card in deck){

swap card with card in random index in deck

}

END

**Class: CardsDriver**

Operation:Main()

START

Deck=new Deck()

OUTPUT Deck

Deck.Shuffle()

DeckCopy=new Deck(Deck)

OUTPUT DeckCopy

Hand1=new Hand(7)

FOR(slots in Hand1){

Hand1.AddACard(DeckCopy.DealACard())

}

FOR(slots in Hand2){

Hand2.AddACard(DeckCopy.DealACard())

}

OUTPUT Hand1

OUTPUT Hand2

END

Test Cases

This specific project actually accepts no input, operations should be tested individually, and the main method can be tested by running it repeatedly, as it is randomized on each run.