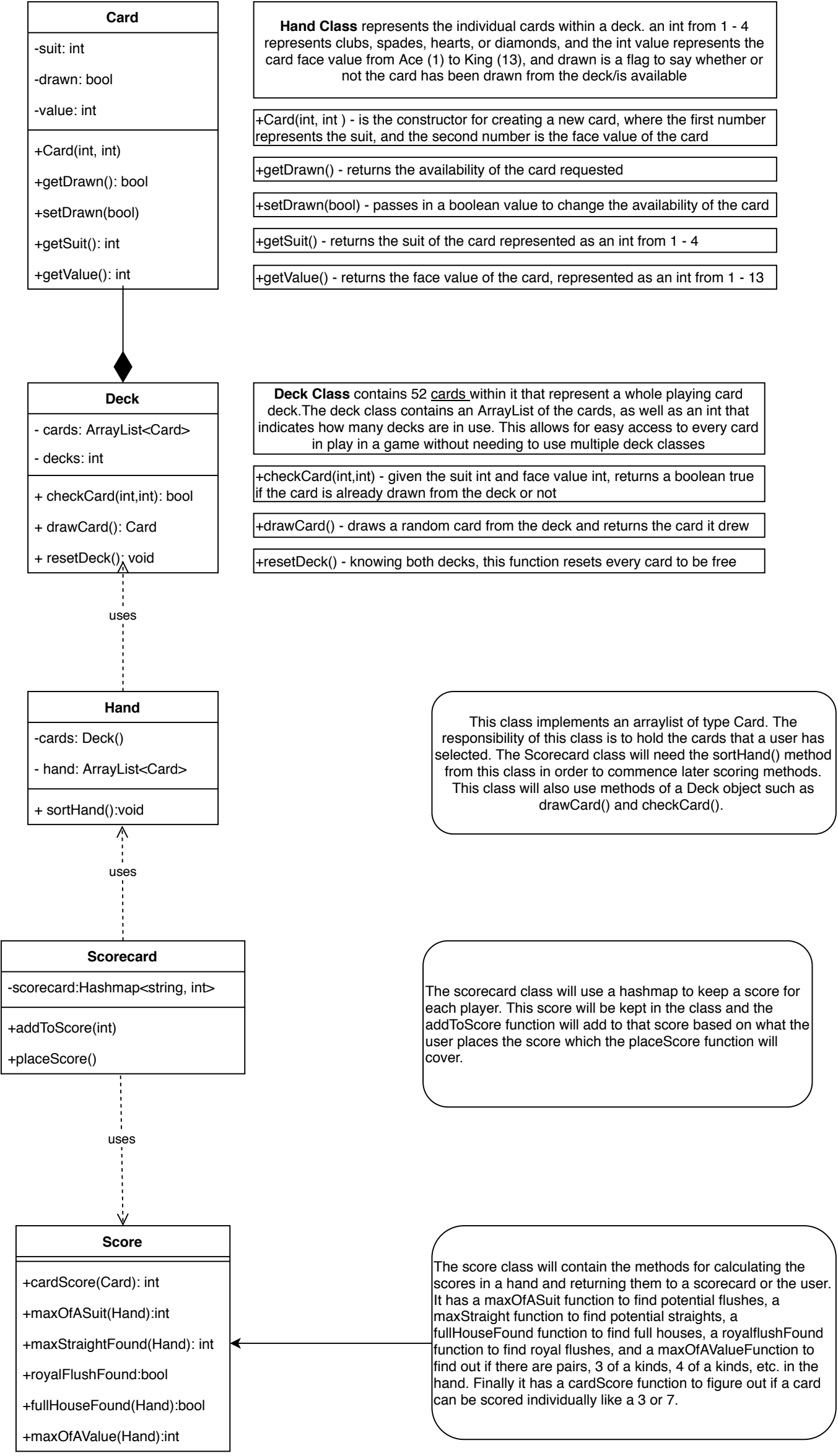
Card
-suit: int
-drawn: bool
-value: int
+Card(int, int)
+getDrawn(): bool
+setDrawn(bool)
+getSuit(): int
+getValue(): int
Hand Class represents the individual cards within a deck. an int from 1 - 4 represents clubs, spades, hearts, or diamonds, and the int value represents the card face value from Ace (1) to King (13), and drawn is a flag to say whether or not the card has been drawn from the deck/is available
+Card(int, int ) - is the constructor for creating a new card, where the first number represents the suit, and the second number is the face value of the card
+getDrawn() - returns the availability of the card requested
+setDrawn(bool) - passes in a boolean value to change the availability of the card
+getSuit() - returns the suit of the card represented as an int from 1 - 4
+getValue() - returns the face value of the card, represented as an int from 1 - 13
Deck
- cards: ArrayList<Card>
- decks: int
+ checkCard(int,int): bool
+ drawCard(): Card
+ resetDeck(): void
Deck Class contains 52 cards within it that represent a whole playing card deck.The deck class contains an ArrayList of the cards, as well as an int that indicates how many decks are in use. This allows for easy access to every card in play in a game without needing to use multiple deck classes
+checkCard(int,int) - given the suit int and face value int, returns a boolean true if the card is already drawn from the deck or not
+drawCard() - draws a random card from the deck and returns the card it drew
+resetDeck() - knowing both decks, this function resets every card to be free
uses
Hand
-cards: Deck()
- hand: ArrayList<Card>
+ sortHand():void
This class implements an arraylist of type Card. The responsibility of this class is to hold the cards that a user has selected. The Scorecard class will need the sortHand() method from this class in order to commence later scoring methods. This class will also use methods of a Deck object such as drawCard() and checkCard().
uses
Scorecard
-scorecard:Hashmap<string, int>
+addToScore(int)
+placeScore()
The scorecard class will use a hashmap to keep a score for each player. This score will be kept in the class and the addToScore function will add to that score based on what the user places the score which the placeScore function will cover.
uses
Score
+cardScore(Card): int
+maxOfASuit(Hand):int
+maxStraightFound(Hand): int
+royalFlushFound:bool
+fullHouseFound(Hand):bool
+maxOfAValue(Hand):int
The score class will contain the methods for calculating the scores in a hand and returning them to a scorecard or the user. It has a maxOfASuit function to find potential flushes, a maxStraight function to find potential straights, a fullHouseFound function to find full houses, a royalflushFound function to find royal flushes, and a maxOfAValueFunction to find out if there are pairs, 3 of a kinds, 4 of a kinds, etc. in the hand. Finally it has a cardScore function to figure out if a card can be scored individually like a 3 or 7.