



## Make11 Design Summary



---

The implementation is built around the following five classes plus additional testing classes.

### Card

Card class represents a playing card in the game make11, we give it predefined attributes such as ranks and suits which is needed for the game to generate a card. The class encapsulates the private rank and suit attributes of a card, making them only accessible through getter methods.

The mentioned getter method **getRank()** and **getSuit()** play a crucial role in generating a card for the user and is utilized by the **Deck** class.

The class also implements a **getRankValue()** method which gets the rank of a specific card, this plays a role in the functionality of the program in comparing a user's card to the computers card.

### Deck

Deck class represents a deck of 52 cards which we generate using the **Card** class and its attributes. This class allows us to give the user a set number of cards and return null when the deck is empty. Furthermore, **deckIsEmpty()** returns true when deck returns null which aids the coding workflow.

The **deal()** method plays a particularly important role in make11 allowing the user to receive a random card and allowing them to remove it from the deck of 52 cards.

The **toString()** method is important in displaying a string representation of the randomly generated card to the user, which helps them progress through the game.

### Highscore

Highscore class provides access to the highscore through getter and increment methods. The class encapsulates the **score** attribute using the *private* modifier.

The mentioned getter method **getScore()** returns the current value of the score attribute.

Alternatively the **increment()** method allows the score to be incremented by 1 when the user scores a point.

## RoundCount

RoundCount class provides access to the current round through getter and increment methods. The class encapsulates the **count** attribute using the *private* modifier.

The mentioned getter method **getCount()** returns the current round of the game.

Alternatively, the **increment()** method allows the round to be incremented by 1 every round as the user progresses.

## Make11

Make11 class contains the **main()** method that runs the game and other methods which assist in the runtime of the program.

Outside the main we are provided with highscore classes which deal with writing and displaying the highscores to the user. **highscoreTable()** - **printScores()** - **addScores()** - **getLowestScore()**

**dealInitialCards()** deals five cards into an array and allows the returned array to be manipulated by replacing a chosen card in the array.

Additionally we have classes that deal with the replay after the user finishes the game, **writeRound()** writes each round to the file, **clearReplay()** clears the replay after the very end of the game. **viewReplay()** prints the replay to the user.

## Test Classes

Test classes have been provided in the following classes: **CardTest**, **DeckTest**, **make11Test**, **HighscoreTest**, **RoundCountTest**.

Test classes test the key individual functionalities of the program which play an important role throughout the course of the game.

The relationship between the classes is as follows. The **Card** class forms the foundation of the individual playing cards, the **Deck** class utilizes the attributes of the card class to form a deck of 52 cards. The main method in **make11** uses the deck to generate five cards to the user while decreasing the deck. Make11 also utilizes Highscore and RoundCount classes throughout the game to keep track of the score and rounds.

