# Assignment 4

Generated by Doxygen 1.8.11

# Contents

# Chapter 1

# Hierarchical Index

## 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

# Chapter 2

# Class Index

## 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

# Chapter 3

# File Index

## 3.1 File List

Here is a list of all documented files with brief descriptions:

# Chapter 4

# Class Documentation

## 4.1 Card Class Reference

Class representing a playing card.

```
#include <CardADT.h>
```

### Public Member Functions

- Card (Rank rank, Suit suit)

  *Constructor for the Card data type.*
- Card ()

  *creating constructor without parameter to dodge initialization*
- Colour getColour ()

  *returns the colour of the card*
- Rank getRank ()

  *returns rank of the card object*
- Suit getSuit ()

  *returns suit of card*

### 4.1.1 Detailed Description

Class representing a playing card.

of a specific suit and rank. Colour is determined by suit

### 4.1.2 Constructor & Destructor Documentation

#### 4.1.2.1 Card::Card ( **Rank** *rank,* **Suit** *suit* )

Constructor for the Card data type.

**Parameters**

| | |
|---|---|
| *rank* | rank or value of card |
| *suit* | suit of card, one of the four suits |

### 4.1.3 Member Function Documentation

#### 4.1.3.1 Colour Card::getColour ( )

returns the colour of the card

**Returns**

red or black depending on suit

#### 4.1.3.2 Rank Card::getRank ( )

returns rank of the card object

**Returns**

rank of card

#### 4.1.3.3 Suit Card::getSuit ( )

returns suit of card

**Returns**

suit of the card

The documentation for this class was generated from the following file:

- include/CardADT.h

## 4.2 Foundation Class Reference

Class representing a Foundation pile in freecell.

```
#include <Foundation.h>
```

## Public Member Functions

- Foundation (Suit suit)

  *Constructor for a single foundation pile.*
- void addCard (Card c)

  *method for adding card onto foundation pile*
- void removeCard ()

  *method for removing top card from foundation pile*
- bool isFull ()

  *checks if foundation pile is complete*
- Suit getSuit ()

  *gets the suit of the foundation*
- Card topCard ()

  *gets the top card from the pile*

### 4.2.1 Detailed Description

Class representing a Foundation pile in freecell.

that accepts a specific suit of cards

### 4.2.2 Constructor & Destructor Documentation

#### 4.2.2.1 Foundation::Foundation ( Suit *suit* )

Constructor for a single foundation pile.

**Parameters**

| | |
|---|---|
| *suit* | suit of the cards belonging to the pile |

### 4.2.3 Member Function Documentation

#### 4.2.3.1 void Foundation::addCard ( Card *c* )

method for adding card onto foundation pile

**Parameters**

| | |
|---|---|
| *c* | card being added to the pile |

#### 4.2.3.2 bool Foundation::isFull ( )

checks if foundation pile is complete

**Returns**

true if full, false otherwise

The documentation for this class was generated from the following file:

- include/Foundation.h

## 4.3 FreeCell Class Reference

Class representing the free cells in a FreeCell game.

```
#include <FreeCell.h>
```

### Public Member Functions

- FreeCell ()

  *constructor for FreeCell*
- bool isFull ()

  *returns true if full, otehrwise false*
- Card searchCard (Rank rank, Suit suit)

  *search for the card with the matching suit and rank*
- void addCard (Card c)

  *move card back to tableau or foundation*
- void removeCard (Rank rank, Suit suit)

  *remove card from free cells*

### 4.3.1 Detailed Description

Class representing the free cells in a FreeCell game.

### 4.3.2 Member Function Documentation

#### 4.3.2.1 void FreeCell::addCard ( Card *c* )

move card back to tableau or foundation

**Parameters**

| | |
|---|---|
| *c* | card being added to free cells pile |

#### 4.3.2.2 bool FreeCell::isFull ( )

returns true if full, otehrwise false

**Returns**

true if all four cells are taken

#### 4.3.2.3 void FreeCell::removeCard ( Rank *rank*, Suit *suit* )

remove card from free cells

**Parameters**

| | |
|---|---|
| *rank* | rank of card being moved |
| *suit* | suit of card being moved |

#### 4.3.2.4 Card FreeCell::searchCard ( Rank *rank*, Suit *suit* )

search for the card with the matching suit and rank

**Parameters**

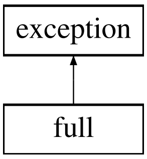
| | |
|---|---|
| *rank* | rank of card being moved |
| *suit* | suit of card being moved |

The documentation for this class was generated from the following file:

- include/FreeCell.h

## 4.4 full Class Reference

Inheritance diagram for full:

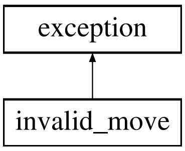


The documentation for this class was generated from the following file:

- include/Exceptions.h

## 4.5 invalid_move Class Reference

Inheritance diagram for invalid_move:

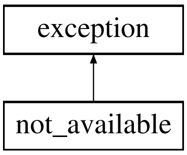


The documentation for this class was generated from the following file:

- include/Exceptions.h

## 4.6 not_available Class Reference

Inheritance diagram for not_available:



The documentation for this class was generated from the following file:

- include/Exceptions.h

## 4.7 Setup Class Reference

### Public Member Functions

- Setup ()
- Setup (vector< vector< Card > > s)
- void freeToTab (Rank rank, Suit suit, int to)

  *moves card from the free cells to a teableau pile*
- void tabToTab (int from, int to)

  *moves card from the Tableau "from" to the tableau "to"*
- void tabToFound (int from)

  *moves card from the Tableau "from" to the foundation*
- void tabToFree (int from)

  *moves card from the Tableau "from" to a free cell*
- void freeToFound (Rank rank, Suit suit)

  *moves card from the free cells to foundation pile*
- void foundToTab (Suit suit, int to)

  *moves top card from a foundation pile to tableau pile*
- void tabTopCards ()

  *returns top cards on the tableau*
- bool **winningGame** ()

## Public Attributes

- Tableau board [8]

    *8 tableau piles*
- Foundation founds [4]

    *4 foundation piles*
- FreeCell free

    *4 free cells*

### 4.7.1 Constructor & Destructor Documentation

#### 4.7.1.1 Setup::Setup ( )

Randomly constructs the initial state of the tableau piles and initializes freecells and foundation piles

#### 4.7.1.2 Setup::Setup ( vector< vector< Card > > *s* )

For manual set up of cards into tableaus

### 4.7.2 Member Function Documentation

#### 4.7.2.1 void Setup::foundToTab ( Suit *suit,* int *to* )

moves top card from a foundation pile to tableau pile

**Parameters**

| | |
|---|---|
| *suit* | suit of card being moved |
| *to* | pile to which it is being moved |

#### 4.7.2.2 void Setup::freeToFound ( Rank *rank,* Suit *suit* )

moves card from the free cells to foundation pile

**Parameters**

| | |
|---|---|
| *rank* | rank of card being moved |
| *suit* | suit of card being moved |

#### 4.7.2.3 void Setup::freeToTab ( Rank *rank,* Suit *suit,* int *to* )

moves card from the free cells to a teableau pile

**Parameters**

| | |
|---|---|
| *rank* | rank of card being moved |
| *suit* | suit of card being moved |
| *to* | index of tableau pile beng moved to |

#### 4.7.2.4 void Setup::tabToFound ( int *from* )

moves card from the Tableau "from" to the foundation

**Parameters**

| | |
|---|---|
| *from* | pile from which a card is being moved to a free cell |

#### 4.7.2.5 void Setup::tabToFree ( int *from* )

moves card from the Tableau "from" to a free cell

**Parameters**

| | |
|---|---|
| *from* | pile from which a card is being moved to foundation |

#### 4.7.2.6 void Setup::tabToTab ( int *from,* int *to* )

moves card from the Tableau "from" to the tableau "to"

**Parameters**

| | |
|---|---|
| *from* | pile from which a card is being moved |
| *to* | pile where card is dropped or appended |

The documentation for this class was generated from the following file:

- include/Setup.h

## 4.8 Tableau Class Reference

Class representing a Tableau pile in freecell.

```
#include <Tableau.h>
```

## Public Member Functions

- Tableau (vector< Card > s)

  *Constructor for Tableau.*
- void addCard (Card c)

  *Adds card c to the tableau pile at hand.*
- void removeCard ()

  *Removes top card from the current pile.*
- Card topCard ()

  *Extracts top card from pile.*

### 4.8.1 Detailed Description

Class representing a Tableau pile in freecell.

where each card

### 4.8.2 Constructor & Destructor Documentation

#### 4.8.2.1 Tableau::Tableau ( vector< **Card** > ***s*** )

Constructor for Tableau.

creates a tableau based on an input vector of cards

**Parameters**

| | |
|---|---|
| *s* | a vector of 6 or 7 cards |

### 4.8.3 Member Function Documentation

#### 4.8.3.1 void Tableau::addCard ( **Card** ***c*** )

Adds card c to the tableau pile at hand.

**Parameters**

| | |
|---|---|
| *c* | card being added to the pile |

#### 4.8.3.2 Card Tableau::topCard ( )

Extracts top card from pile.

**Returns**

Card object with the attributes of the card

The documentation for this class was generated from the following file:

- include/Tableau.h

# Chapter 5

# File Documentation

## 5.1 include/CardADT.h File Reference

Representing a playing card as an ADT.

### Classes

- class Card

  *Class representing a playing card.*

### Enumerations

- enum Suit { **C**, **D**, **S**, **H** }

  *enum type for card suits in the order Clubs, Diamonds, spades and hearts respectively*
- enum Colour { **Red**, **Black** }

  *enum type for card suit colour*
- enum Rank {
  **Empty**, **ace**, **two**, **three**,
  **four**, **five**, **six**, **seven**,
  **eight**, **nine**, **ten**, **jack**,
  **queen**, **king** }

  *enum type for card rank or value*

### 5.1.1 Detailed Description

Representing a playing card as an ADT.

along with enume types for card attributes

## 5.2 include/Foundation.h File Reference

Representing one of four foundation piles.

```
#include <vector>
```

### Classes

- class Foundation

    *Class representing a Foundation pile in freecell.*

### 5.2.1 Detailed Description

Representing one of four foundation piles.

one for each suit

## 5.3 include/FreeCell.h File Reference

Representing one of four free cells.

```
#include <vector>
```

### Classes

- class FreeCell

    *Class representing the free cells in a FreeCell game.*

### 5.3.1 Detailed Description

Representing one of four free cells.

## 5.4 include/Setup.h File Reference

Representing the setup of the board and methods.

```
#include <vector>
```

### Classes

- class Setup

### 5.4.1 Detailed Description

Representing the setup of the board and methods.

for changing the game state

## 5.5 include/Tableau.h File Reference

Representing one of eight tableau piles.

```
#include <vector>
```

### Classes

- class Tableau

  *Class representing a Tableau pile in freecell.*

### 5.5.1 Detailed Description

Representing one of eight tableau piles.