

# Mastermind

1.0

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>Class Documentation</b>	<b>3</b>
2.1	Mastermind Class Reference . . . . .	3
2.1.1	Detailed Description . . . . .	3
2.1.2	Constructor & Destructor Documentation . . . . .	3
2.1.2.1	Mastermind() . . . . .	3
2.2	Pattern Class Reference . . . . .	4
2.2.1	Detailed Description . . . . .	4
2.2.2	Constructor & Destructor Documentation . . . . .	4
2.2.2.1	Pattern() . . . . .	4
2.2.3	Member Function Documentation . . . . .	5
2.2.3.1	compareTo() . . . . .	5
2.2.3.2	getPegColor() . . . . .	5
2.2.3.3	len() . . . . .	5
2.2.3.4	randomize() . . . . .	6
2.2.3.5	setPegColor() . . . . .	6
2.3	Score Class Reference . . . . .	6
2.3.1	Detailed Description . . . . .	7
2.3.2	Constructor & Destructor Documentation . . . . .	7
2.3.2.1	Score() . . . . .	7
2.3.3	Member Function Documentation . . . . .	7
2.3.3.1	getNumBlack() . . . . .	7

2.3.3.2	<a href="#">getNumWhite()</a>	7
2.4	<a href="#">TextInput Class Reference</a>	8
2.4.1	<a href="#">Detailed Description</a>	8
2.4.2	<a href="#">Constructor &amp; Destructor Documentation</a>	8
2.4.2.1	<a href="#">TextInput()</a>	8
2.4.3	<a href="#">Member Function Documentation</a>	8
2.4.3.1	<a href="#">enterGuess()</a>	9
2.4.3.2	<a href="#">queryLengthOfPattern()</a>	9
2.4.3.3	<a href="#">queryNewGame()</a>	9
2.4.3.4	<a href="#">queryNumberOfColors()</a>	9
2.4.3.5	<a href="#">queryNumberOfTurns()</a>	10
2.5	<a href="#">TextOutput Class Reference</a>	10
2.5.1	<a href="#">Detailed Description</a>	10
2.5.2	<a href="#">Constructor &amp; Destructor Documentation</a>	10
2.5.2.1	<a href="#">TextOutput()</a>	10

# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">Mastermind</a>	Main class for the <a href="#">Mastermind</a> game . . . . .	3
<a href="#">Pattern</a>	Class for storing a color pattern for <a href="#">Mastermind</a> . . . . .	4
<a href="#">Score</a>	A score for a single turn from game of <a href="#">Mastermind</a> . . . . .	6
<a href="#">TextInput</a>	Class for dealing with text-based input for the <a href="#">Mastermind</a> game . . . . .	8
<a href="#">TextOutput</a>	Provide text-based output for the <a href="#">Mastermind</a> game . . . . .	10



## Chapter 2

# Class Documentation

### 2.1 Mastermind Class Reference

Main class for the [Mastermind](#) game.

```
#include <Mastermind.h>
```

#### Public Member Functions

- [Mastermind](#) ([TextInput](#) &inputManager, [TextOutput](#) &outputManager)  
*Create a new instance of the [Mastermind](#) game.*

#### 2.1.1 Detailed Description

Main class for the [Mastermind](#) game.

#### 2.1.2 Constructor & Destructor Documentation

##### 2.1.2.1 Mastermind()

```
Mastermind::Mastermind (  
    TextInput & inputManager,  
    TextOutput & outputManager )
```

Create a new instance of the [Mastermind](#) game.

#### Parameters

<i>inputManager</i>	instance of class that gathers input from the user
<i>outputManager</i>	instance of class that displays output to the user

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

- /Users/Victor/Desktop/Cpp-Refresher/Cpp-Project-Exercise/master\_mind/Mastermind.h
- /Users/Victor/Desktop/Cpp-Refresher/Cpp-Project-Exercise/master\_mind/Mastermind.cpp

## 2.2 Pattern Class Reference

Class for storing a color pattern for [Mastermind](#).

```
#include <Pattern.h>
```

### Public Member Functions

- [Pattern](#) (const int numPegs)  
*Construct a new pattern.*
- int [len](#) () const  
*Return the length of the current pattern.*
- int [getPegColor](#) (const int index) const  
*Return the current color setting (an integer) of the specified peg.*
- void [setPegColor](#) (const int index, const int color)  
*Set the color of a peg at the given index of the pattern.*
- [Score compareTo](#) (const [Pattern](#) &otherPattern) const  
*Compare the current pattern to another and calculate the score.*
- void [randomize](#) (const int numColors)  
*Make a random pattern.*

### 2.2.1 Detailed Description

Class for storing a color pattern for [Mastermind](#).

### 2.2.2 Constructor & Destructor Documentation

#### 2.2.2.1 Pattern()

```
Pattern::Pattern (  
    const int numPegs )
```

Construct a new pattern.

Initially, the pattern consists of numPegs pegs, each set to color 0.



## Parameters

<i>numPegs</i>	the length of the pattern
----------------	---------------------------

## 2.2.3 Member Function Documentation

### 2.2.3.1 compareTo()

```
Score Pattern::compareTo (
    const Pattern & otherPattern ) const
```

Compare the current pattern to another and calculate the score.

## Parameters

<i>otherPattern</i>	the pattern to be compared to the current one
---------------------	---

## Returns

a [Score](#) instance representing the result.

### 2.2.3.2 getPegColor()

```
int Pattern::getPegColor (
    const int index ) const
```

Return the current color setting (an integer) of the specified peg.

## Parameters

<i>index</i>	the index of the peg
--------------	----------------------

## Returns

the peg's color

### 2.2.3.3 len()

```
int Pattern::len ( ) const
```

Return the length of the current pattern.

**Returns**

the length of the pattern

**2.2.3.4 randomize()**

```
void Pattern::randomize (
    const int numColors )
```

Make a random pattern.

**Parameters**

<i>numColors</i>	the maximum number of colors to use in the pattern
------------------	--

**2.2.3.5 setPegColor()**

```
void Pattern::setPegColor (
    const int index,
    const int color )
```

Set the color of a peg at the given index of the pattern.

**Parameters**

<i>index</i>	the index of the peg
<i>colorID</i>	the desired color identifier (an integer)

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

- /Users/Victor/Desktop/Cpp-Refresher/Cpp-Project-Exercise/master\_mind/Pattern.h
- /Users/Victor/Desktop/Cpp-Refresher/Cpp-Project-Exercise/master\_mind/Pattern.cpp

**2.3 Score Class Reference**

A score for a single turn from game of [Mastermind](#).

```
#include <Score.h>
```

**Public Member Functions**

- [Score](#) (const int numBlack, const int numWhite)  
*Create score with given black and white components.*
- int [getNumBlack](#) () const  
*Get the black component of the score.*
- int [getNumWhite](#) () const  
*Get the white component of the score.*

### 2.3.1 Detailed Description

A score for a single turn from game of [Mastermind](#).

A "black" component designates the number of pegs that are exact matches for the answer. A "white" component counts pegs that are correctly colored but not well positioned.

### 2.3.2 Constructor & Destructor Documentation

#### 2.3.2.1 Score()

```
Score::Score (
    const int numBlack,
    const int numWhite ) [inline]
```

Create score with given black and white components.

##### Parameters

<i>numBlack</i>	the black component of the score
<i>numWhite</i>	the white component of the score

### 2.3.3 Member Function Documentation

#### 2.3.3.1 getNumBlack()

```
int Score::getNumBlack ( ) const [inline]
```

Get the black component of the score.

##### Returns

the number of pegs scored as black

#### 2.3.3.2 getNumWhite()

```
int Score::getNumWhite ( ) const [inline]
```

Get the white component of the score.

##### Returns

the number of pegs scored as white

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

- `/Users/Victor/Desktop/Cpp-Refresher/Cpp-Project-Exercise/master_mind/Score.h`

## 2.4 TextInput Class Reference

Class for dealing with text-based input for the [Mastermind](#) game.

```
#include <TextInput.h>
```

### Public Member Functions

- [TextInput](#) (const vector< string > &colorNames)  
*Create a new text input instance.*
- int [queryLengthOfPattern](#) ()  
*Ask the user how many pegs in the secret pattern.*
- int [queryNumberOfColors](#) ()  
*Ask the user how many colors to use for secret pattern.*
- int [queryNumberOfTurns](#) () const  
*Ask the user maximum number of guesses to be allowed.*
- bool [queryNewGame](#) () const  
*Offer the user a new game.*
- [Pattern](#) [enterGuess](#) () const  
*Get a guess from the user and return it as a [Pattern](#) instance.*

### 2.4.1 Detailed Description

Class for dealing with text-based input for the [Mastermind](#) game.

### 2.4.2 Constructor & Destructor Documentation

#### 2.4.2.1 TextInput()

```
TextInput::TextInput (
    const vector< string > & colorNames )
```

Create a new text input instance.

#### Parameters

<i>colorNames</i>	a list of strings (each color must start with a different letter)
-------------------	---

### 2.4.3 Member Function Documentation

### 2.4.3.1 enterGuess()

```
Pattern TextInput::enterGuess ( ) const
```

Get a guess from the user and return it as a [Pattern](#) instance.

#### Returns

the pattern entered

### 2.4.3.2 queryLengthOfPattern()

```
int TextInput::queryLengthOfPattern ( )
```

Ask the user how many pegs in the secret pattern.

The length of the pattern is also stored internally

#### Returns

the length of the pattern

### 2.4.3.3 queryNewGame()

```
bool TextInput::queryNewGame ( ) const
```

Offer the user a new game.

#### Returns

true if accepted, false otherwise

### 2.4.3.4 queryNumberOfColors()

```
int TextInput::queryNumberOfColors ( )
```

Ask the user how many colors to use for secret pattern.

The number of colors is also stored internally.

#### Returns

the number of colors

### 2.4.3.5 queryNumberOfTurns()

```
int TextInput::queryNumberOfTurns ( ) const
```

Ask the user maximum number of guesses to be allowed.

#### Returns

the maximum number of guesses

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

- /Users/Victor/Desktop/Cpp-Refresher/Cpp-Project-Exercise/master\_mind/TextInput.h
- /Users/Victor/Desktop/Cpp-Refresher/Cpp-Project-Exercise/master\_mind/TextInput.cpp

## 2.5 TextOutput Class Reference

Provide text-based output for the [Mastermind](#) game.

```
#include <TextOutput.h>
```

### Public Member Functions

- [TextOutput](#) (const vector< string > &colorNames)  
*Construct a new [TextOutput](#) instance.*
- void [startGame](#) (int lengthOfPattern, int maxNumberOfTurns)  
*Game is beginning with specified parameters.*
- void [displayTurn](#) (const [Pattern](#) &guess, const [Score](#) &result)  
*Display recent guess [Pattern](#) and resulting [Score](#) to the screen.*
- void [announceVictory](#) (const [Pattern](#) &secret) const  
*Inform the player that he/she has correctly matched the secret [Pattern](#).*
- void [announceDefeat](#) (const [Pattern](#) &secret) const  
*Inform the player that he/she has lost and reveal the secret [Pattern](#).*

### 2.5.1 Detailed Description

Provide text-based output for the [Mastermind](#) game.

### 2.5.2 Constructor & Destructor Documentation

#### 2.5.2.1 TextOutput()

```
TextOutput::TextOutput (
    const vector< string > & colorNames )
```

Construct a new [TextOutput](#) instance.

## Parameters

<i>colorNames</i>	a sequence of strings (each color must start with a different letter)
-------------------	---

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

- /Users/Victor/Desktop/Cpp-Refresher/Cpp-Project-Exercise/master\_mind/TextOutput.h
- /Users/Victor/Desktop/Cpp-Refresher/Cpp-Project-Exercise/master\_mind/TextOutput.cpp

