Scaling and Conversion System

1.0

Generated by Doxygen 1.8.18

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 files with brief descriptions:

Chapter 4

Class Documentation

4.1 Fraction::BadInput Class Reference

`#include <Fraction.h>`

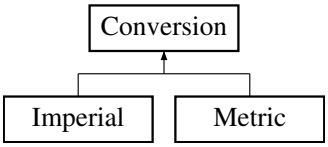
4.1.1 Detailed Description

@Exeption class for bad input

4.2 Conversion Class Reference

`#include <Conversion.h>`

Inheritance diagram for Conversion:



Classes

- class NegativeInput

Public Member Functions

- Conversion ()

  default Constructor, Conversion class
- Conversion (int value)

  integer constructor, Conversion class
- Conversion (double value)

  double constructor, Conversion class
- double getValue ()

Protected Member Functions

- void setValue (double val)

  modifier for attribute value, Conversion class

Protected Attributes

- double value

4.2.1 Constructor & Destructor Documentation

4.2.1.1 Conversion() [1/3]

```
Conversion::Conversion ( )
```

default Constructor, Conversion class

Parameters

none	

Returns

none

4.2.1.2 Conversion() [2/3]

```
Conversion::Conversion (
            int value )
```

integer constructor, Conversion class

Parameters

integer	value

Returns

none

4.2.1.3 Conversion() [3/3]

```
Conversion::Conversion (
            double value )
```

double constructor, Conversion class

Parameters

none	

Returns

none

4.2.2 Member Function Documentation

4.2.2.1 getValue()

```
double Conversion::getValue ( )
```

4.2.2.2 setValue()

```
void Conversion::setValue (
            double val )  [protected]
```

modifier for attribute value, Conversion class

Parameters

none	

Returns

void

Exceptions

NegativeInput()	{ exception thown if negative input detected for conversion. }

4.2.3 Member Data Documentation

4.2.3.1 value

```
double Conversion::value  [protected]
```

4.3 Fraction::DivisionByZero Class Reference

Exception class for a division by zero.

`#include <Fraction.h>`

4.3.1 Detailed Description

Exception class for a division by zero.

4.4 Fraction Class Reference

`#include <Fraction.h>`

Classes

- class BadInput
- class DivisionByZero

  Exception class for a division by zero.

Public Member Functions

- Fraction ()

  default Constructor, Fraction Class
- Fraction (int value)

  integer constructor, Fraction Class
- Fraction (double value)

  double constructor, Fraction Class
- Fraction (double num, double denum)

  double, double constructor, Fraction Class
- Fraction (const Fraction &frac)

  Default Constructor Fraction Class.
- double getNumerator ()

  accesor the current value of numerator
- double getDenominator ()

  accesor to the current value of denominator
- Fraction operator+ (Fraction const &frac)

  Overload of plus operator for Fraction class.
- Fraction operator- (Fraction const &frac)

  Overload of minus operator for Fraction class.
- Fraction operator/ (Fraction const &frac)

  Overload of division operator for Fraction class.
- Fraction operator∗ (Fraction const &frac)

  Overload of multiplication operator for Fraction class.

Friends

- std::istream & operator>> (std::istream &in, Fraction &frac)

  Overload of operator >> for Fraction class.
- std::ostream & operator<< (std::ostream &os, const Fraction &frac)

  Overload of operator << for Fraction class.

4.4.1 Constructor & Destructor Documentation

4.4.1.1 Fraction() [1/5]

```
Fraction::Fraction ( )
```

default Constructor, Fraction Class

Parameters

none	

Returns

none

4.4.1.2 Fraction() [2/5]

```
Fraction::Fraction (
            int number )
```

integer constructor, Fraction Class

Parameters

integer	number

Returns

none

4.4.1.3 Fraction() [3/5]

```
Fraction::Fraction (
            double numerator_ )
```

double constructor, Fraction Class

Parameters

double	number

Returns

none

4.4.1.4 Fraction() [4/5]

```
Fraction::Fraction (
            double numerator_,
            double denominator_ )
```

double, double constructor, Fraction Class

Parameters

double	nomerator, double denominator of the fraction

Returns

none

4.4.1.5 Fraction() [5/5]

```
Fraction::Fraction (
            const Fraction & frac )
```

Default Constructor Fraction Class.

Parameters

none	

Returns

none

4.4.2 Member Function Documentation

4.4.2.1 getDenominator()

```
double Fraction::getDenominator ( )
```

accesor to the current value of denominator

Parameters

none	

Returns

double denominator

4.4.2.2 getNumerator()

```
double Fraction::getNumerator ( )
```

accesor the current value of numerator

Parameters

none	

Returns

double numerator

4.4.2.3 operator∗()

```
Fraction Fraction::operator* (
            Fraction const & frac )
```

Overload of multiplication operator for Fraction class.

Parameters

constant	reference to a Fraction object

Returns

Fraction object

4.4.2.4 operator+()

```
Fraction Fraction::operator+ (
            Fraction const & frac )
```

Overload of plus operator for Fraction class.

Parameters

constant	reference to a Fraction object

Returns

Fraction object

4.4.2.5 operator-()

```
Fraction Fraction::operator- (
            Fraction const & frac )
```

Overload of minus operator for Fraction class.

Parameters

constant	reference to a Fraction object

Returns

Fraction object

4.4.2.6 operator/()

```
Fraction Fraction::operator/ (
            Fraction const & frac )
```

Overload of division operator for Fraction class.

Parameters

constant	reference to a Fraction object

Returns

Fraction object

4.4.3 Friends And Related Function Documentation

4.4.3.1 operator<<

```
std::ostream& operator<< (
            std::ostream & os,
            const Fraction & frac )  [friend]
```

Overload of operator << for Fraction class.

Parameters

reference	to an ostream object , constant reference to a Fraction object

Returns

reference to an ostream object

4.4.3.2 operator>>

```
std::istream& operator>> (
            std::istream & in,
            Fraction & frac )  [friend]
```

Overload of operator >> for Fraction class.

Parameters

reference	to an istream object , reference to a Fraction object

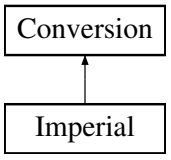
Returns

reference to an istream object

4.5 Imperial Class Reference

```
#include <Imperial.h>
```

Inheritance diagram for Imperial:



Public Member Functions

- Imperial ()
  default Constructor, Imperial Class
- Imperial (int value)
  integer Constructor, Metric Class
- Imperial (double value)
  double Constructor, Metric Class
- double milligramsToOunce ()
  conversion milligrams to ounces
- double gramsToOunce ()
  conversion grams to ounces
- double kilogramsToPounds ()
  conversion kilograms to pounds
- double millimetersToInch ()
  conversion millimeters to inches
- double centimetersToInch ()
  conversion centimeters to inches
- double metersToInch ()
  conversion meters to inches
- double metersToFeet ()
  conversion meters to feet
- double metersToYard ()
  conversion meters to yards
- double kilometersToYards ()
  conversion kilometers to yards
- double kilometersToMiles ()
  conversion kilometers to miles

Additional Inherited Members

4.5.1 Constructor & Destructor Documentation

4.5.1.1 Imperial() [1/3]

```
Imperial::Imperial ( )
```

default Constructor, Imperial Class

Parameters

none	

Returns

none

4.5.1.2 Imperial() [2/3]

```
Imperial::Imperial (
            int value )
```

integer Constructor, Metric Class

Parameters

integer	value

Returns

none

4.5.1.3 Imperial() [3/3]

```
Imperial::Imperial (
            double value )
```

double Constructor, Metric Class

Parameters

double	value

Returns

none

4.5.2 Member Function Documentation

4.5.2.1 centimetersToInch()

```
double Imperial::centimetersToInch ( )
```

conversion centimeters to inches

Parameters

none	

Returns

double result;

4.5.2.2 gramsToOunce()

```
double Imperial::gramsToOunce ( )
```

conversion grams to ounces

Parameters

none	

Returns

double result;

4.5.2.3 kilogramsToPounds()

```
double Imperial::kilogramsToPounds ( )
```

conversion kilograms to pounds

Parameters

none	

Returns

double result;

4.5.2.4 kilometersToMiles()

```
double Imperial::kilometersToMiles ( )
```

conversion kilometers to miles

Parameters

none	

Returns

double result;

4.5.2.5 kilometersToYards()

```
double Imperial::kilometersToYards ( )
```

conversion kilometers to yards

Parameters

none	

Returns

double result;

4.5.2.6 metersToFeet()

```
double Imperial::metersToFeet ( )
```

conversion meters to feet

Parameters

none	

Returns

double result;

4.5.2.7 metersToInch()

```
double Imperial::metersToInch ( )
```

conversion meters to inches

Parameters

none	

Returns

double result;

4.5.2.8 metersToYard()

```
double Imperial::metersToYard ( )
```

conversion meters to yards

Parameters

none	

Returns

double result;

4.5.2.9 milligramsToOunce()

```
double Imperial::milligramsToOunce ( )
```

conversion milligrams to ounces

Parameters

none	

Returns

double result;

4.5.2.10 millimetersToInch()

```
double Imperial::millimetersToInch ( )
```

conversion millimeters to inches

Parameters

none	

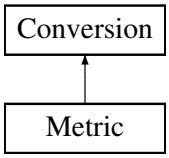
Returns

double result;

4.6 Metric Class Reference

`#include <Metric.h>`

Inheritance diagram for Metric:



Public Member Functions

- Metric ()
  default Constructor, Metric Class
- Metric (int value)
  integer Constructor, Metric Class
- Metric (double value)
  double Constructor, Metric Class
- double ounceToMilligrams ()
  conversion ounces to milligrams
- double ounceToGrams ()
  conversion ounces to grams
- double poundsToKilograms ()
  conversion pounds to kilograms
- double inchToMillimeters ()
  conversion inches to millimeters
- double inchToCentimeters ()
  conversion inches to centimeters
- double inchToMeters ()
  conversion inches to meters
- double feetToMeters ()
  conversion feet to meters
- double yardsToMeters ()
  conversion yards to meters
- double yardsToKilometers ()
  conversion yards to kilometers
- double milesToKilometers ()
  conversion miles to kilometers

Additional Inherited Members

4.6.1 Constructor & Destructor Documentation

4.6.1.1 Metric() [1/3]

```
Metric::Metric ( )
```

default Constructor, Metric Class

Parameters

none	

Returns

none

4.6.1.2 Metric() [2/3]

```
Metric::Metric (
            int value )
```

integer Constructor, Metric Class

Parameters

integer	value

Returns

none

4.6.1.3 Metric() [3/3]

```
Metric::Metric (
            double value )
```

double Constructor, Metric Class

Parameters

double	value

Returns

none

4.6.2 Member Function Documentation

4.6.2.1 feetToMeters()

```
double Metric::feetToMeters ( )
```

conversion feet to meters

Parameters

none	

Returns

double result;

4.6.2.2 inchToCentimeters()

```
double Metric::inchToCentimeters ( )
```

conversion inches to centimeters

Parameters

none	

Returns

double result;

4.6.2.3 inchToMeters()

```
double Metric::inchToMeters ( )
```

conversion inches to meters

Parameters

none	

Returns

double result;

4.6.2.4 inchToMillimeters()

```
double Metric::inchToMillimeters ( )
```

conversion inches to millimeters

Parameters

none	

Returns

double result;

4.6.2.5 milesToKilometers()

```
double Metric::milesToKilometers ( )
```

conversion miles to kilometers

Parameters

none	

Returns

double result;

4.6.2.6 ounceToGrams()

```
double Metric::ounceToGrams ( )
```

conversion ounces to grams

Parameters

none	

Returns

double result;

4.6.2.7 ounceToMilligrams()

```
double Metric::ounceToMilligrams ( )
```

conversion ounces to milligrams

Parameters

none	

Returns

double result;

4.6.2.8 poundsToKilograms()

```
double Metric::poundsToKilograms ( )
```

conversion pounds to kilograms

Parameters

none	

Returns

double result;

4.6.2.9 yardsToKilometers()

```
double Metric::yardsToKilometers ( )
```

conversion yards to kilometers

Parameters

none	

Returns

double result;

4.6.2.10 yardsToMeters()

```
double Metric::yardsToMeters ( )
```

conversion yards to meters

Parameters

none	

Returns

double result;

4.7 Conversion::NegativeInput Class Reference

```
#include <Conversion.h>
```

4.7.1 Detailed Description

@Exception class for a negative input

Chapter 5

File Documentation

5.1 Conversion.cpp File Reference

Conversion class definition for conversion of units.

```
#include "Conversion.h"
```

5.1.1 Detailed Description

Conversion class definition for conversion of units.

Version

1.0

Date

04/28/2020

Author

Frederick Vitug

Mohammed Mazous

Robert Hartnett

Luis Silva @title Conversion Class

5.2 Conversion.h File Reference

Conversion class header for conversion of units.

```
#include <string>
#include <cmath>
```

Classes

- class Conversion
- class Conversion::NegativeInput

5.2.1 Detailed Description

Conversion class header for conversion of units.

Version

1.0

Date

04/28/2020

Author

Frederick Vitug

Mohammed Mazous

Robert Hartnett

Luis Silva @title Conversion Class

5.3 Fraction.cpp File Reference

```
#include "Fraction.h"
```

Functions

- std::istream & operator>> (std::istream &in, Fraction &frac)

  Overload of operator >> for Fraction class.
- std::ostream & operator<< (std::ostream &os, const Fraction &frac)

  Overload of operator << for Fraction class.

5.3.1 Function Documentation

5.3.1.1 operator<<()

```
std::ostream& operator<< (
            std::ostream & os,
            const Fraction & frac )
```

Overload of operator << for Fraction class.

Parameters

reference	to an ostream object , constant reference to a Fraction object

Returns

reference to an ostream object

5.3.1.2 operator>>()

```
std::istream& operator>> (
            std::istream & in,
            Fraction & frac )
```

Overload of operator >> for Fraction class.

Parameters

reference	to an istream object , reference to a Fraction object

Returns

reference to an istream object

5.4 Fraction.h File Reference

```
#include <iostream>
#include <cmath>
#include <string>
#include <stdlib.h>
#include <algorithm>
```

Classes

- class Fraction
- class Fraction::DivisionByZero

  Exception class for a division by zero.
- class Fraction::BadInput

Macros

- #define FRACTION_H

5.4.1 Macro Definition Documentation

5.4.1.1 FRACTION_H

```
#define FRACTION_H
```

5.5 Imperial.cpp File Reference

Imperial class definition for conversion Imperial - Metric units.

```
#include "Imperial.h"
```

5.5.1 Detailed Description

Imperial class definition for conversion Imperial - Metric units.

Version

1.0

Date

04/28/2020

Author

Frederick Vitug

Mohammed Mazous

Robert Hartnett

Luis Silva @title Imperial Class

5.6 Imperial.h File Reference

Imperial class header for conversion Imperial - Metric units.

```
#include "Conversion.h"
```

Classes

- class Imperial

5.6.1 Detailed Description

Imperial class header for conversion Imperial - Metric units.

Version

1.0

Date

04/28/2020

Author

Frederick Vitug

Mohammed Mazous

Robert Hartnett

Luis Silva @title Imperial Class

5.7 main.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include "Fraction.h"
#include "Imperial.h"
#include "Metric.h"
```

Functions

- void displayMenu ()
- void launchOption (int option)
- void scalingFraction ()
- void displayConversionMenu ()
- void conversion ()
- void metricConverter (int option, double value)
- void imperialConverter (int option, double value)
- int main ()
- Fraction readFraction ()

5.7.1 Function Documentation

5.7.1.1 conversion()

```
void conversion ( )
```

5.7.1.2 displayConversionMenu()

```
void displayConversionMenu ( )
```

5.7.1.3 displayMenu()

```
void displayMenu ( )
```

5.7.1.4 imperialConverter()

```
void imperialConverter (
            int option,
            double value )
```

5.7.1.5 launchOption()

```
void launchOption (
            int option )
```

5.7.1.6 main()

```
int main ( )
```

5.7.1.7 metricConverter()

```
void metricConverter (
            int option,
            double value )
```

5.7.1.8 readFraction()

```
Fraction readFraction ( )
```

5.7.1.9 scalingFraction()

```
void scalingFraction ( )
```

5.8 Metric.cpp File Reference

Metric class definition for conversion Metric - Imperial units.

```
#include "Metric.h"
```

5.8.1 Detailed Description

Metric class definition for conversion Metric - Imperial units.

Version

1.0

Date

04/28/2020

Author

Frederick Vitug

Mohammed Mazous

Robert Hartnett

Luis Silva @title Metric Class

5.9 Metric.h File Reference

Metric class header for conversion Metric-Imperial units.

```
#include "Conversion.h"
```

Classes

- class Metric

5.9.1 Detailed Description

Metric class header for conversion Metric-Imperial units.

Version

1.0

Date

04/28/2020

Author

Frederick Vitug

Mohammed Mazous

Robert Hartnett

Luis Silva @title Metric Class

Index