

Scaling and Conversion System

1.0

Generated by Doxygen 1.8.18

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 Fraction::BadInput Class Reference	7
4.1.1 Detailed Description	7
4.2 Conversion Class Reference	7
4.2.1 Constructor & Destructor Documentation	8
4.2.1.1 Conversion() [1/3]	8
4.2.1.2 Conversion() [2/3]	8
4.2.1.3 Conversion() [3/3]	9
4.2.2 Member Function Documentation	9
4.2.2.1 getValue()	9
4.2.2.2 setValue()	9
4.2.3 Member Data Documentation	10
4.2.3.1 value	10
4.3 Fraction::DivisionByZero Class Reference	10
4.3.1 Detailed Description	10
4.4 Fraction Class Reference	10
4.4.1 Constructor & Destructor Documentation	11
4.4.1.1 Fraction() [1/5]	11
4.4.1.2 Fraction() [2/5]	11
4.4.1.3 Fraction() [3/5]	11
4.4.1.4 Fraction() [4/5]	13
4.4.1.5 Fraction() [5/5]	13
4.4.2 Member Function Documentation	13
4.4.2.1 getDenominator()	14
4.4.2.2 getNumerator()	14
4.4.2.3 operator*()	14
4.4.2.4 operator+()	15
4.4.2.5 operator-()	15
4.4.2.6 operator/()	15
4.4.3 Friends And Related Function Documentation	16
4.4.3.1 operator<<	16
4.4.3.2 operator>>	16
4.5 Imperial Class Reference	16
4.5.1 Constructor & Destructor Documentation	17

4.5.1.1 Imperial() [1/3]	17
4.5.1.2 Imperial() [2/3]	18
4.5.1.3 Imperial() [3/3]	18
4.5.2 Member Function Documentation	18
4.5.2.1 centimetersToInch()	18
4.5.2.2 gramsToOunce()	19
4.5.2.3 kilogramsToPounds()	19
4.5.2.4 kilometersToMiles()	19
4.5.2.5 kilometersToYards()	20
4.5.2.6 metersToFeet()	20
4.5.2.7 metersToInch()	20
4.5.2.8 metersToYard()	21
4.5.2.9 milligramsToOunce()	21
4.5.2.10 millimetersToInch()	21
4.6 Metric Class Reference	22
4.6.1 Constructor & Destructor Documentation	22
4.6.1.1 Metric() [1/3]	23
4.6.1.2 Metric() [2/3]	23
4.6.1.3 Metric() [3/3]	23
4.6.2 Member Function Documentation	24
4.6.2.1 feetToMeters()	24
4.6.2.2 inchToCentimeters()	24
4.6.2.3 inchToMeters()	24
4.6.2.4 inchToMillimeters()	25
4.6.2.5 milesToKilometers()	25
4.6.2.6 ounceToGrams()	25
4.6.2.7 ounceToMilligrams()	26
4.6.2.8 poundsToKilograms()	26
4.6.2.9 yardsToKilometers()	26
4.6.2.10 yardsToMeters()	27
4.7 Conversion::NegativeInput Class Reference	27
4.7.1 Detailed Description	27
5 File Documentation	29
5.1 Conversion.cpp File Reference	29
5.1.1 Detailed Description	29
5.2 Conversion.h File Reference	29
5.2.1 Detailed Description	30
5.3 Fraction.cpp File Reference	30
5.3.1 Function Documentation	30
5.3.1.1 operator<<()	30
5.3.1.2 operator>>()	31

5.4 Fraction.h File Reference	31
5.4.1 Macro Definition Documentation	32
5.4.1.1 FRACTION_H	32
5.5 Imperial.cpp File Reference	32
5.5.1 Detailed Description	32
5.6 Imperial.h File Reference	32
5.6.1 Detailed Description	33
5.7 main.cpp File Reference	33
5.7.1 Function Documentation	33
5.7.1.1 conversion()	33
5.7.1.2 displayConversionMenu()	34
5.7.1.3 displayMenu()	34
5.7.1.4 imperialConverter()	34
5.7.1.5 launchOption()	34
5.7.1.6 main()	34
5.7.1.7 metricConverter()	34
5.7.1.8 readFraction()	34
5.7.1.9 scalingFraction()	35
5.8 Metric.cpp File Reference	35
5.8.1 Detailed Description	35
5.9 Metric.h File Reference	35
5.9.1 Detailed Description	36
Index	37

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Fraction::BadInput	7
Conversion	7
Imperial	16
Metric	22
Fraction::DivisionByZero	10
Fraction	10
Conversion::NegativeInput	27

Chapter 2

Class Index

2.1 Class List

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

Fraction::BadInput	7
Conversion	7
Fraction::DivisionByZero	
Exception class for a division by zero	10
Fraction	10
Imperial	16
Metric	22
Conversion::NegativeInput	27

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Conversion.cpp	
Conversion	class definition for conversion of units 29
Conversion.h	
Conversion	class header for conversion of units 29
Fraction.cpp 30
Fraction.h 31
Imperial.cpp	
Imperial	class definition for conversion Imperial - Metric units 32
Imperial.h	
Imperial	class header for conversion Imperial - Metric units 32
main.cpp 33
Metric.cpp	
Metric	class definition for conversion Metric - Imperial units 35
Metric.h	
Metric	class header for conversion Metric-Imperial units 35

Chapter 4

Class Documentation

4.1 Fraction::BadInput Class Reference

```
#include <Fraction.h>
```

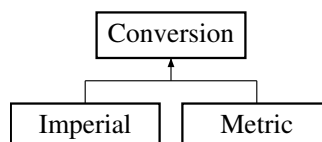
4.1.1 Detailed Description

@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

<i>none</i>	
-------------	--

Returns

none

4.2.1.2 [Conversion\(\)](#) [2/3]

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

integer constructor, [Conversion](#) class

Parameters

<i>integer</i>	value
----------------	-------

Returns

none

4.2.1.3 Conversion() [3/3]

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

double constructor, [Conversion](#) class

Parameters

<i>none</i>	
-------------	--

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

<i>none</i>	
-------------	--

Returns

void

Exceptions

<i>NegativeInput()</i>	{ exception thrown 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

<i>none</i>	
-------------	--

Returns

none

4.4.1.2 [Fraction\(\)](#) [2/5]

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

integer constructor, [Fraction](#) Class

Parameters

<i>integer</i>	<i>number</i>
----------------	---------------

Returns

none

4.4.1.3 [Fraction\(\)](#) [3/5]

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

double constructor, [Fraction](#) Class

Parameters

<i>double</i>	number
---------------	--------

Returns

none

4.4.1.4 Fraction() [4/5]

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

double, double constructor, [Fraction](#) Class

Parameters

<i>double</i>	numerator, double denominator of the fraction
---------------	---

Returns

none

4.4.1.5 Fraction() [5/5]

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

Default Constructor [Fraction](#) Class.

Parameters

<i>none</i>	
-------------	--

Returns

none

4.4.2 Member Function Documentation

4.4.2.1 getDenominator()

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

accesor to the current value of denominator

Parameters

<i>none</i>	
-------------	--

Returns

double denominator

4.4.2.2 getNumerator()

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

accesor the current value of numerator

Parameters

<i>none</i>	
-------------	--

Returns

double numerator

4.4.2.3 operator*()

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

Overload of multiplication operator for [Fraction](#) class.

Parameters

<i>constant</i>	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

<i>constant</i>	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

<i>constant</i>	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

<i>constant</i>	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

<i>reference</i>	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

<i>reference</i>	to an istream object , reference to a Fraction object
------------------	---

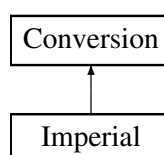
Returns

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

<i>none</i>	
-------------	--

Returns

none

4.5.1.2 Imperial() [2/3]

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

integer Constructor, [Metric](#) Class

Parameters

<i>integer</i>	value
----------------	-------

Returns

none

4.5.1.3 Imperial() [3/3]

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

double Constructor, [Metric](#) Class

Parameters

<i>double</i>	value
---------------	-------

Returns

none

4.5.2 Member Function Documentation

4.5.2.1 centimetersToInch()

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

conversion centimeters to inches

Parameters

<i>none</i>	
-------------	--

Returns

double result;

4.5.2.2 gramsToOunce()

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

conversion grams to ounces

Parameters

<i>none</i>	
-------------	--

Returns

double result;

4.5.2.3 kilogramsToPounds()

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

conversion kilograms to pounds

Parameters

<i>none</i>	
-------------	--

Returns

double result;

4.5.2.4 kilometersToMiles()

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

conversion kilometers to miles

Parameters

<i>none</i>	
-------------	--

Returns

double result;

4.5.2.5 kilometersToYards()

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

conversion kilometers to yards

Parameters

<i>none</i>	
-------------	--

Returns

double result;

4.5.2.6 metersToFeet()

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

conversion meters to feet

Parameters

<i>none</i>	
-------------	--

Returns

double result;

4.5.2.7 metersToInch()

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

conversion meters to inches

Parameters

<i>none</i>	
-------------	--

Returns

double result;

4.5.2.8 metersToYard()

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

conversion meters to yards

Parameters

<i>none</i>	
-------------	--

Returns

double result;

4.5.2.9 milligramsToOunce()

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

conversion milligrams to ounces

Parameters

<i>none</i>	
-------------	--

Returns

double result;

4.5.2.10 millimetersToInch()

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

conversion millimeters to inches

Parameters

<i>none</i>	
-------------	--

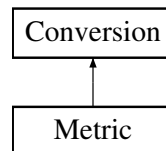
Returns

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

<i>none</i>	
-------------	--

Returns

none

4.6.1.2 Metric() [2/3]

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

integer Constructor, [Metric Class](#)

Parameters

<i>integer</i>	value
----------------	-------

Returns

none

4.6.1.3 Metric() [3/3]

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

double Constructor, [Metric Class](#)

Parameters

<i>double</i>	value
---------------	-------

Returns

none

4.6.2 Member Function Documentation

4.6.2.1 feetToMeters()

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

conversion feet to meters

Parameters

<i>none</i>	
-------------	--

Returns

double result;

4.6.2.2 inchToCentimeters()

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

conversion inches to centimeters

Parameters

<i>none</i>	
-------------	--

Returns

double result;

4.6.2.3 inchToMeters()

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

conversion inches to meters

Parameters

<i>none</i>	
-------------	--

Returns

double result;

4.6.2.4 inchToMillimeters()

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

conversion inches to millimeters

Parameters

<i>none</i>	
-------------	--

Returns

double result;

4.6.2.5 milesToKilometers()

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

conversion miles to kilometers

Parameters

<i>none</i>	
-------------	--

Returns

double result;

4.6.2.6 ounceToGrams()

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

conversion ounces to grams

Parameters

<i>none</i>	
-------------	--

Returns

double result;

4.6.2.7 ounceToMilligrams()

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

conversion ounces to milligrams

Parameters

<i>none</i>	
-------------	--

Returns

double result;

4.6.2.8 poundsToKilograms()

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

conversion pounds to kilograms

Parameters

<i>none</i>	
-------------	--

Returns

double result;

4.6.2.9 yardsToKilometers()

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

conversion yards to kilometers

Parameters

<i>none</i>	
-------------	--

Returns

double result;

4.6.2.10 yardsToMeters()

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

conversion yards to meters

Parameters

<i>none</i>	
-------------	--

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

<i>reference</i>	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

<i>reference</i>	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

- centimetersToInch
 - Imperial, [18](#)
- Conversion, [7](#)
 - Conversion, [8](#)
 - getValue, [9](#)
 - setValue, [9](#)
 - value, [10](#)
- conversion
 - main.cpp, [33](#)
- Conversion.cpp, [29](#)
- Conversion.h, [29](#)
- Conversion::NegativeInput, [27](#)
- displayConversionMenu
 - main.cpp, [33](#)
- displayMenu
 - main.cpp, [34](#)
- feetToMeters
 - Metric, [24](#)
- Fraction, [10](#)
 - Fraction, [11](#), [13](#)
 - getDenominator, [13](#)
 - getNumerator, [14](#)
 - operator<<, [16](#)
 - operator>>, [16](#)
 - operator*, [14](#)
 - operator+, [14](#)
 - operator-, [15](#)
 - operator/, [15](#)
- Fraction.cpp, [30](#)
 - operator<<, [30](#)
 - operator>>, [31](#)
- Fraction.h, [31](#)
 - FRACTION_H, [32](#)
- Fraction::BadInput, [7](#)
- Fraction::DivisionByZero, [10](#)
- FRACTION_H
 - Fraction.h, [32](#)
- getDenominator
 - Fraction, [13](#)
- getNumerator
 - Fraction, [14](#)
- getValue
 - Conversion, [9](#)
- gramsToOunce
 - Imperial, [19](#)
- Imperial, [16](#)
 - centimetersToInch, [18](#)
 - gramsToOunce, [19](#)
 - Imperial, [17](#), [18](#)
 - kilogramsToPounds, [19](#)
 - kilometersToMiles, [19](#)
 - kilometersToYards, [20](#)
 - metersToFeet, [20](#)
 - metersToInch, [20](#)
 - metersToYard, [21](#)
 - milligramsToOunce, [21](#)
 - millimetersToInch, [21](#)
- Imperial.cpp, [32](#)
- Imperial.h, [32](#)
- imperialConverter
 - main.cpp, [34](#)
- inchToCentimeters
 - Metric, [24](#)
- inchToMeters
 - Metric, [24](#)
- inchToMillimeters
 - Metric, [25](#)
- kilogramsToPounds
 - Imperial, [19](#)
- kilometersToMiles
 - Imperial, [19](#)
- kilometersToYards
 - Imperial, [20](#)
- launchOption
 - main.cpp, [34](#)
- main
 - main.cpp, [34](#)
- main.cpp, [33](#)
 - conversion, [33](#)
 - displayConversionMenu, [33](#)
 - displayMenu, [34](#)
 - imperialConverter, [34](#)
 - launchOption, [34](#)
 - main, [34](#)
 - metricConverter, [34](#)
 - readFraction, [34](#)
 - scalingFraction, [34](#)
- metersToFeet
 - Imperial, [20](#)
- metersToInch
 - Imperial, [20](#)
- metersToYard
 - Imperial, [21](#)

- Metric, [22](#)
 - feetToMeters, [24](#)
 - inchToCentimeters, [24](#)
 - inchToMeters, [24](#)
 - inchToMillimeters, [25](#)
 - Metric, [22](#), [23](#)
 - milesToKilometers, [25](#)
 - ounceToGrams, [25](#)
 - ounceToMilligrams, [26](#)
 - poundsToKilograms, [26](#)
 - yardsToKilometers, [26](#)
 - yardsToMeters, [27](#)
- Metric.cpp, [35](#)
- Metric.h, [35](#)
- metricConverter
 - main.cpp, [34](#)
- milesToKilometers
 - Metric, [25](#)
- milligramsToOunce
 - Imperial, [21](#)
- millimetersToInch
 - Imperial, [21](#)
- operator<<
 - Fraction, [16](#)
 - Fraction.cpp, [30](#)
- operator>>
 - Fraction, [16](#)
 - Fraction.cpp, [31](#)
- operator*
 - Fraction, [14](#)
- operator+
 - Fraction, [14](#)
- operator-
 - Fraction, [15](#)
- operator/
 - Fraction, [15](#)
- ounceToGrams
 - Metric, [25](#)
- ounceToMilligrams
 - Metric, [26](#)
- poundsToKilograms
 - Metric, [26](#)
- readFraction
 - main.cpp, [34](#)
- scalingFraction
 - main.cpp, [34](#)
- setValue
 - Conversion, [9](#)
- value
 - Conversion, [10](#)
- yardsToKilometers
 - Metric, [26](#)
- yardsToMeters
 - Metric, [27](#)