

LongInt

Generated by Doxygen 1.8.20



<b>1 Class Index</b>	<b>1</b>
1.1 Class List	1
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 LongInt Class Reference	5
3.1.1 Constructor & Destructor Documentation	7
3.1.1.1 LongInt() [1/2]	7
3.1.1.2 LongInt() [2/2]	7
3.1.2 Member Function Documentation	7
3.1.2.1 operator*() [1/2]	7
3.1.2.2 operator*() [2/2]	8
3.1.2.3 operator*=( ) [1/2]	8
3.1.2.4 operator*=( ) [2/2]	8
3.1.2.5 operator+( ) [1/2]	8
3.1.2.6 operator+( ) [2/2]	9
3.1.2.7 operator++( ) [1/2]	9
3.1.2.8 operator++( ) [2/2]	9
3.1.2.9 operator+=( ) [1/2]	9
3.1.2.10 operator+=( ) [2/2]	10
3.1.2.11 operator-( ) [1/2]	10
3.1.2.12 operator-( ) [2/2]	10
3.1.2.13 operator--( ) [1/2]	10
3.1.2.14 operator--( ) [2/2]	11
3.1.2.15 operator-=( ) [1/2]	11
3.1.2.16 operator-=( ) [2/2]	11
3.1.2.17 operator/( ) [1/2]	11
3.1.2.18 operator/( ) [2/2]	11
3.1.2.19 operator/=( ) [1/2]	12
3.1.2.20 operator/=( ) [2/2]	12
3.1.2.21 operator<( ) [1/2]	12
3.1.2.22 operator<( ) [2/2]	12
3.1.2.23 operator<=( ) [1/2]	13
3.1.2.24 operator<=( ) [2/2]	13
3.1.2.25 operator=( ) [1/3]	13
3.1.2.26 operator=( ) [2/3]	13
3.1.2.27 operator=( ) [3/3]	13
3.1.2.28 operator==( ) [1/2]	14
3.1.2.29 operator==( ) [2/2]	14
3.1.2.30 operator>( ) [1/2]	14
3.1.2.31 operator>( ) [2/2]	14

3.1.2.32 operator>=() [1/2]	15
3.1.2.33 operator>=() [2/2]	15
3.1.3 Friends And Related Function Documentation	15
3.1.3.1 division_by_two	15
3.1.3.2 operator<<	15
3.1.3.3 operator>>	16
3.1.4 Member Data Documentation	16
3.1.4.1 minus	16
3.1.4.2 number	16
<b>4 File Documentation</b>	<b>17</b>
4.1 input.txt File Reference	17
4.2 LongInt.cpp File Reference	17
4.2.1 Function Documentation	17
4.2.1.1 division_by_two()	17
4.2.1.2 increment_minus()	18
4.2.1.3 increment_plus()	18
4.2.1.4 operator<<()	18
4.2.1.5 operator>>()	18
4.3 LongInt.h File Reference	18
4.3.1 Function Documentation	19
4.3.1.1 increment_minus()	19
4.3.1.2 increment_plus()	19
4.4 main.cpp File Reference	19
4.4.1 Function Documentation	19
4.4.1.1 main()	19
<b>Index</b>	<b>21</b>

# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">LongInt</a> . . . . .	5
-----------------------------------	---



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">LongInt.cpp</a>	.....	17
<a href="#">LongInt.h</a>	.....	18
<a href="#">main.cpp</a>	.....	19





## Chapter 3

# Class Documentation

### 3.1 LongInt Class Reference

```
#include <LongInt.h>
```

#### Public Member Functions

- [LongInt \(\)](#)  
*Default constructor.*
- [LongInt \(const int num\)](#)  
*Constructor that sets the number.*
- [LongInt operator+ \(const LongInt &other\)](#)  
*Returns the sum of two LongInt variables.*
- [LongInt operator+ \(int number\)](#)  
*Returns the sum of LongInt and int variables.*
- [LongInt operator+= \(const LongInt &other\)](#)  
*Assigns to a LongInt variable the sum of itself and other LongInt variable.*
- [LongInt operator+= \(int number\)](#)  
*Assigns to a LongInt variable the sum of itself and int variable.*
- [LongInt operator- \(const LongInt &other\)](#)  
*Returns the subtraction of LongInt by LongInt variable.*
- [LongInt operator- \(int number\)](#)  
*Returns the subtraction of LongInt by int variable.*
- [LongInt operator-= \(const LongInt &other\)](#)  
*Assigns to a LongInt variable the subtraction of itself by other LongInt variable.*
- [LongInt operator-= \(int number\)](#)  
*Assigns to a LongInt variable the subtraction of itself by int variable.*
- [LongInt operator\\* \(const LongInt &other\)](#)  
*Returns the multiplication of LongInt and LongInt variable.*
- [LongInt operator\\* \(const int number\)](#)  
*Returns the multiplication of LongInt and int variable.*
- [LongInt operator\\*= \(const LongInt &other\)](#)  
*Assigns to a LongInt variable the multiplication of itself and other LongInt variable.*
- [LongInt operator\\*= \(const int number\)](#)  
*Assigns to a LongInt variable the multiplication of itself and int variable.*

- `LongInt operator/` (const int &other)  
Returns the division of `LongInt` by int variable.
- `LongInt operator/=` (const int &other)  
Assigns to a `LongInt` variable the division of itself by int variable.
- `LongInt operator/` (`LongInt` &other)  
Returns the division of `LongInt` by `LongInt` variable.
- `LongInt operator/=` (`LongInt` &other)  
Assigns to a `LongInt` variable the division of itself by other `LongInt` variable.
- `LongInt operator++` (int notused)  
Increases `LongInt` variable by one after the action(postfix notation)
- `LongInt operator++` ()  
Increases `LongInt` variable by one before the action(prefix notation)
- `LongInt operator--` (int notused)  
Decreases `LongInt` variable by one after the action(postfix notation)
- `LongInt operator--` ()  
Decreases `LongInt` variable by one before the action(prefix notation)
- `LongInt & operator=` (int &value)  
Assigns `LongInt` virable int's value.
- `LongInt & operator=` (const int &value)  
Assigns `LongInt` virable const int's value.
- `LongInt & operator=` (`LongInt` obj)  
Assigns `LongInt` virable `LongInt` value.
- `bool operator==` (`LongInt` &other)  
Returns true if two `LongInt` variables are equal, else returns false.
- `bool operator==` (int number)  
Returns true if `LongInt` and int variables are equal, else returns false.
- `bool operator<` (const `LongInt` &other)  
Returns true if left `LongInt` virable is lesser than right `LongInt` variable.
- `bool operator<` (const int &number)  
Returns true if left `LongInt` virable is lesser than right const int variable.
- `bool operator<=` (`LongInt` &other)  
Returns true if left `LongInt` virable is lesser or equals right `LongInt` variable.
- `bool operator<=` (const int &number)  
Returns true if left `LongInt` virable is lesser or equals right const int variable.
- `bool operator>` (const `LongInt` &other)  
Returns true if left `LongInt` virable is more than right const `LongInt` variable.
- `bool operator>` (const int &number)  
Returns true if left `LongInt` virable is more than right const int variable.
- `bool operator>=` (`LongInt` &other)  
Returns true if left `LongInt` virable is more or equals right const `LongInt` variable.
- `bool operator>=` (const int &number)  
Returns true if left `LongInt` virable is more or equals right const int variable.

## Private Attributes

- `vector< int >` number
- `bool` minus

## Friends

- ostream & [operator<<](#) (ostream &out, [LongInt](#) obj)  
*Input stream.*
- istream & [operator>>](#) (istream &in, [LongInt](#) &obj)  
*Output stream.*
- [LongInt division\\_by\\_two](#) ([LongInt](#) z)  
*Sub-function, returns [LongInt](#) variable divided by two.*

## 3.1.1 Constructor & Destructor Documentation

### 3.1.1.1 [LongInt\(\)](#) [1/2]

```
LongInt::LongInt ( )
```

Default constructor.

### 3.1.1.2 [LongInt\(\)](#) [2/2]

```
LongInt::LongInt (
    const int num )
```

Constructor that sets the number.

## 3.1.2 Member Function Documentation

### 3.1.2.1 [operator\\*\(\)](#) [1/2]

```
LongInt LongInt::operator* (
    const int number )
```

Returns the multiplication of [LongInt](#) and int variable.

#### Returns

the multiplication of [LongInt](#) and int variable

### 3.1.2.2 operator\*() [2/2]

```
LongInt LongInt::operator* (
    const LongInt & other )
```

Returns the multiplication of [LongInt](#) and [LongInt](#) variable.

Returns the multiplication of [LongInt](#) and [LongInt](#) variable

### 3.1.2.3 operator\*=( ) [1/2]

```
LongInt LongInt::operator*= (
    const int number )
```

Assigns to a [LongInt](#) variable the multiplication of itself and int variable.

### 3.1.2.4 operator\*=( ) [2/2]

```
LongInt LongInt::operator*= (
    const LongInt & other )
```

Assigns to a [LongInt](#) variable the multiplication of itself and other [LongInt](#) variable.

#### Returns

The multiplication of [LongInt](#) variable by other [LongInt](#) variable

### 3.1.2.5 operator+( ) [1/2]

```
LongInt LongInt::operator+ (
    const LongInt & other )
```

Returns the sum of two [LongInt](#) variables.

#### Returns

[LongInt](#) variable that equals the sum of two [LongInt](#) variables

### 3.1.2.6 operator+() [2/2]

```
LongInt LongInt::operator+ (
    int number )
```

Returns the sum of [LongInt](#) and int variables.

#### Returns

[LongInt](#) variable equals the sum of [LongInt](#) and int variables

### 3.1.2.7 operator++() [1/2]

```
LongInt LongInt::operator++ ( )
```

Increases [LongInt](#) variable by one before the action(prefix notation)

#### Returns

Increased [LongInt](#) variable by one before the action(prefix notation)

### 3.1.2.8 operator++() [2/2]

```
LongInt LongInt::operator++ (
    int notused )
```

Increases [LongInt](#) variable by one after the action(postfix notation)

#### Returns

Increased [LongInt](#) variable by one after the action(postfix notation)

### 3.1.2.9 operator+=() [1/2]

```
LongInt LongInt::operator+= (
    const LongInt & other )
```

Assigns to a [LongInt](#) variable the sum of itself and other [LongInt](#) variable.

#### Returns

Assigns to a [LongInt](#) variable the sum of itself and other [LongInt](#) variable

### 3.1.2.10 operator+=() [2/2]

```
LongInt LongInt::operator+= (
    int number )
```

Assigns to a [LongInt](#) variable the sum of itself and int variable.

Assigns to a [LongInt](#) variable the sum of itself and int variable

### 3.1.2.11 operator-() [1/2]

```
LongInt LongInt::operator- (
    const LongInt & other )
```

Returns the subtraction of [LongInt](#) by [LongInt](#) variable.

#### Returns

the subtraction of [LongInt](#) by [LongInt](#) variable

### 3.1.2.12 operator-() [2/2]

```
LongInt LongInt::operator- (
    int number )
```

Returns the subtraction of [LongInt](#) by int variable.

#### Returns

the subtraction of [LongInt](#) by int variable

### 3.1.2.13 operator--() [1/2]

```
LongInt LongInt::operator-- ( )
```

Decreases [LongInt](#) variable by one before the action(prefix notation)

#### Returns

Decreased [LongInt](#) variable by one before the action(prefix notation)

**3.1.2.14 operator--()** [2/2]

```
LongInt LongInt::operator-- (
    int notused )
```

Decreases [LongInt](#) variable by one after the action(postfix notation)

**Returns**

Decreased [LongInt](#) variable by one after the action(postfix notation)

**3.1.2.15 operator-=()** [1/2]

```
LongInt LongInt::operator-= (
    const LongInt & other )
```

Assigns to a [LongInt](#) variable the subtraction of itself by other [LongInt](#) variable.

Assigns to a [LongInt](#) variable the subtraction of itself by other [LongInt](#) variable

**3.1.2.16 operator-=()** [2/2]

```
LongInt LongInt::operator-= (
    int number )
```

Assigns to a [LongInt](#) variable the subtraction of itself by int variable.

Assigns to a [LongInt](#) variable the subtraction of itself by int variable

**3.1.2.17 operator/()** [1/2]

```
LongInt LongInt::operator/ (
    const int & other )
```

Returns the division of [LongInt](#) by int variable.

**Returns**

The division of [LongInt](#) by int variable

**3.1.2.18 operator/()** [2/2]

```
LongInt LongInt::operator/ (
    LongInt & other )
```

Returns the division of [LongInt](#) by [LongInt](#) variable.

### 3.1.2.19 operator/=( ) [1/2]

```
LongInt LongInt::operator/= (
    const int & other )
```

Assigns to a [LongInt](#) variable the division of itself by int variable.

### 3.1.2.20 operator/=( ) [2/2]

```
LongInt LongInt::operator/= (
    LongInt & other )
```

Assigns to a [LongInt](#) variable the division of itself by other [LongInt](#) variable.

#### Returns

A [LongInt](#) variable divided by other [LongInt](#) variable

### 3.1.2.21 operator<( ) [1/2]

```
bool LongInt::operator< (
    const int & number )
```

Returns true if left [LongInt](#) virable is lesser than right const int variable.

#### Returns

true if left [LongInt](#) virable is lesser than right const int variable, else false

### 3.1.2.22 operator<( ) [2/2]

```
bool LongInt::operator< (
    const LongInt & other )
```

Returns true if left [LongInt](#) virable is lesser than right [LongInt](#) variable.

#### Returns

true if left [LongInt](#) virable is lesser than right [LongInt](#) variable,else false



**3.1.2.23 operator<=()** [1/2]

```
bool LongInt::operator<= (
    const int & number )
```

Returns true if left [LongInt](#) virable is lesser or equals right const int variable.

**Returns**

true if left [LongInt](#) virable is lesser or equals right const int variable, else false

**3.1.2.24 operator<=()** [2/2]

```
bool LongInt::operator<= (
    LongInt & other )
```

Returns true if left [LongInt](#) virable is lesser or equals right [LongInt](#) variable.

**Returns**

true if left [LongInt](#) virable is lesser or equals right [LongInt](#) variable, else false

**3.1.2.25 operator=()** [1/3]

```
LongInt & LongInt::operator= (
    const int & value )
```

Assigs [LongInt](#) virable const int's value.

**3.1.2.26 operator=()** [2/3]

```
LongInt & LongInt::operator= (
    int & value )
```

Assigs [LongInt](#) virable int's value.

**3.1.2.27 operator=()** [3/3]

```
LongInt & LongInt::operator= (
    LongInt obj )
```

Assigs [LongInt](#) virable [LongInt](#) value.

### 3.1.2.28 operator==( ) [1/2]

```
bool LongInt::operator==(
    int number )
```

Returns true if [LongInt](#) and int variables are equal, else returns false.

#### Returns

true if [LongInt](#) and int variables are equal, else false

### 3.1.2.29 operator==( ) [2/2]

```
bool LongInt::operator==(
    LongInt & other )
```

Returns true if two [LongInt](#) variables are equal, else returns false.

#### Returns

true if two [LongInt](#) variables are equal, else returns false

### 3.1.2.30 operator>( ) [1/2]

```
bool LongInt::operator> (
    const int & number )
```

Returns true if left [LongInt](#) virable is more than right const int variable.

#### Returns

true if left [LongInt](#) virable is more than right const int variable else false

### 3.1.2.31 operator>( ) [2/2]

```
bool LongInt::operator> (
    const LongInt & other )
```

Returns true if left [LongInt](#) virable is more than right const [LongInt](#) variable.

#### Returns

true if left [LongInt](#) virable is more than right const [LongInt](#) variable, else false

### 3.1.2.32 operator>=() [1/2]

```
bool LongInt::operator>= (
    const int & number )
```

Returns true if left [LongInt](#) virable is more or equals right const int variable.

#### Returns

true if left [LongInt](#) virable is more or equals right const int variable, else false

### 3.1.2.33 operator>=() [2/2]

```
bool LongInt::operator>= (
    LongInt & other )
```

Returns true if left [LongInt](#) virable is more or equals right const [LongInt](#) variable.

#### Returns

true if left [LongInt](#) virable is more or equals right const [LongInt](#) variable else false

## 3.1.3 Friends And Related Function Documentation

### 3.1.3.1 division\_by\_two

```
LongInt division_by_two (
    LongInt z ) [friend]
```

Sub-function, returns [LongInt](#) variable divided by two.

#### Returns

[LongInt](#) variable divided by two

### 3.1.3.2 operator<<

```
ostream& operator<< (
    ostream & out,
    LongInt obj ) [friend]
```

Input stream.

### 3.1.3.3 operator>>

```
istream& operator>> (
    istream & in,
    LongInt & obj ) [friend]
```

Output stream.

## 3.1.4 Member Data Documentation

### 3.1.4.1 minus

```
bool LongInt::minus [private]
```

### 3.1.4.2 number

```
vector<int> LongInt::number [private]
```

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

- [LongInt.h](#)
- [LongInt.cpp](#)

## Chapter 4

# File Documentation

### 4.1 input.txt File Reference

### 4.2 LongInt.cpp File Reference

```
#include "LongInt.h"
```

#### Functions

- ostream & [operator<<](#) (ostream &out, [LongInt](#) obj)
- istream & [operator>>](#) (istream &in, [LongInt](#) &obj)
- void [increment\\_plus](#) (vector< int > &obj)  
*Increases the number in vector's type by one.*
- void [increment\\_minus](#) (vector< int > &obj)  
*Decreases the number in vector's type by one.*
- [LongInt division\\_by\\_two](#) ([LongInt](#) z)

#### 4.2.1 Function Documentation

##### 4.2.1.1 division\_by\_two()

```
LongInt division_by_two (  
    LongInt z )
```

#### Returns

[LongInt](#) variable divided by two

#### 4.2.1.2 increment\_minus()

```
void increment_minus (
    vector< int > & obj )
```

Decreases the number in vector's type by one.

##### Returns

Increased number in vector's type by one

#### 4.2.1.3 increment\_plus()

```
void increment_plus (
    vector< int > & obj )
```

Increases the number in vector's type by one.

##### Returns

Decreased number in vector's type by one

#### 4.2.1.4 operator<<()

```
ostream& operator<< (
    ostream & out,
    LongInt obj )
```

#### 4.2.1.5 operator>>()

```
istream& operator>> (
    istream & in,
    LongInt & obj )
```

## 4.3 LongInt.h File Reference

```
#include <iostream>
#include <vector>
#include <string>
```

## Classes

- class [LongInt](#)

## Functions

- void [increment\\_plus](#) (vector< int > &obj)  
*Increases the number in vector's type by one.*
- void [increment\\_minus](#) (vector< int > &obj)  
*Decreases the number in vector's type by one.*

### 4.3.1 Function Documentation

#### 4.3.1.1 increment\_minus()

```
void increment_minus (  
    vector< int > & obj )
```

Decreases the number in vector's type by one.

##### Returns

Increased number in vector's type by one

#### 4.3.1.2 increment\_plus()

```
void increment_plus (  
    vector< int > & obj )
```

Increases the number in vector's type by one.

##### Returns

Decreased number in vector's type by one

## 4.4 main.cpp File Reference

```
#include <iostream>  
#include <vector>  
#include <string>  
#include "LongInt.h"
```

## Functions

- int [main](#) ()

### 4.4.1 Function Documentation

#### 4.4.1.1 main()

```
int main ( )
```





# Index

division\_by\_two  
    LongInt, 15  
    LongInt.cpp, 17

increment\_minus  
    LongInt.cpp, 17  
    LongInt.h, 19

increment\_plus  
    LongInt.cpp, 18  
    LongInt.h, 19

input.txt, 17

LongInt, 5  
    division\_by\_two, 15  
    LongInt, 7  
    minus, 16  
    number, 16  
    operator<, 12  
    operator<<, 15  
    operator<=, 12, 13  
    operator>, 14  
    operator>>, 15  
    operator>=, 14, 15  
    operator\*, 7  
    operator\*=: 8  
    operator+, 8  
    operator++, 9  
    operator+=, 9  
    operator-, 10  
    operator--, 10  
    operator=, 11  
    operator/, 11  
    operator/=: 11, 12  
    operator=, 13  
    operator==, 13, 14

LongInt.cpp, 17  
    division\_by\_two, 17  
    increment\_minus, 17  
    increment\_plus, 18  
    operator<<, 18  
    operator>>, 18

LongInt.h, 18  
    increment\_minus, 19  
    increment\_plus, 19

main  
    main.cpp, 19

main.cpp, 19  
    main, 19

minus

LongInt, 16

number  
    LongInt, 16

operator<  
    LongInt, 12

operator<<  
    LongInt, 15  
    LongInt.cpp, 18

operator<=  
    LongInt, 12, 13

operator>  
    LongInt, 14

operator>>  
    LongInt, 15  
    LongInt.cpp, 18

operator>=  
    LongInt, 14, 15

operator\*  
    LongInt, 7

operator\*=  
    LongInt, 8

operator+  
    LongInt, 8

operator++  
    LongInt, 9

operator+=  
    LongInt, 9

operator-  
    LongInt, 10

operator--  
    LongInt, 10

operator-=  
    LongInt, 11

operator/  
    LongInt, 11

operator/=  
    LongInt, 11, 12

operator=  
    LongInt, 13

operator==  
    LongInt, 13, 14