

Thank you for printing our content at www.domain-name.com. Please check back soon for new contents.



3d: 19h: 20m: 21s

Stop copy pasting code you don't actually understand

Build the coding confidence you need to become a developer companies will fight
for

[Become a PRO](#)

(https://programiz.pro/offer/january-2026?utm_source=programiz.com&utm_medium=referral&utm_audience=ORGANIC-FREEMIUM&utm_campaign=newyear-26&utm_content=interests_two_months_free&utm_term=sticky_banner_launch)

[learn/master-cpp?](#)

[.com&utm_medium=referral&utm_audience=ORGANIC-align=course_promotion&utm_content=interests_learn_cpp&utm_term=nav_floating_button](#))

```
double divide_by_zero = 7 / 0;
```

The above code causes an exception as it is not possible to divide a number by **0**.

The process of handling these types of errors in C++ is known as exception handling.

In C++, we handle exceptions with the help of the `try` and `catch` blocks, along with the `throw` keyword.

- `try` - code that may raise an exception
- `throw` - throws an exception when an error is detected
- `catch` - code that handles the exception thrown by the `throw` keyword

Note: The `throw` statement is not compulsory, especially if we use standard C++ exceptions.

Syntax for Exception Handling in C++

The basic syntax for exception handling in C++ is given below:

Thank you for printing our content at www.domain-name.com. Please check back soon for new contents.



3d: 19h: 20m: 21s

Stop copy pasting code you don't actually understand

Build the coding confidence you need to become a developer companies will fight
for

Become a PRO

(https://programiz.pro/offer/january-2026?utm_source=programiz.com&utm_medium=referral&utm_audience=ORGANIC-FREEMIUM&utm_campaign=newyear-26&utm_content=interests_two_months_free&utm_term=sticky_banner_launch)

[learn/master-cpp?](#)

[.com&utm_medium=referral&utm_audience=ORGANIC-sign=course_promotion&utm_content=interests_learn_cpp&utm_term=nav_floating_button](#))

The `catch` block cannot be used without the `try` block.

Thank you for printing our content at www.domain-name.com. Please check back soon for new contents.



3d: 19h: 20m: 21s

Stop copy pasting code you don't actually understand

Build the coding confidence you need to become a developer companies will fight
for

Become a PRO

(https://programiz.pro/offer/january-2026?utm_source=programiz.com&utm_medium=referral&utm_audience=ORGANIC-FREEMIUM&utm_campaign=newyear-26&utm_content=interests_two_months_free&utm_term=sticky_banner_launch)

[learn/master-cpp?](#)

[.com&utm_medium=referral&utm_audience=ORGANIC-align=course_promotion&utm_content=interests_learn_cpp&utm_term=nav_floating_button](#))

```
try {  
  
    // throw an exception if denominator is 0  
    if (denominator == 0)  
        throw 0;  
  
    // not executed if denominator is 0  
    divide = numerator / denominator;  
    cout << numerator << " / " << denominator << " = " << divide << endl;  
}  
  
catch (int num_exception) {  
    cout << "Error: Cannot divide by 0" << endl;  
}
```

[Run Code ➞ \(/cpp-programming/online-compiler\)](#)

Output 1

```
Enter numerator: 72  
Enter denominator: 0  
Error: Cannot divide by 0
```

Output 2

```
Enter numerator: 72  
Enter denominator: 3  
72 / 3 = 24
```

Thank you for printing our content at www.domain-name.com. Please check back soon for new contents.



3d: 19h: 20m: 21s

Stop copy pasting code you don't actually understand

Build the coding confidence you need to become a developer companies will fight
for

Become a PRO

(https://programiz.pro/offer/january-2026?utm_source=programiz.com&utm_medium=referral&utm_audience=ORGANIC-FREEMIUM&utm_campaign=newyear-26&utm_content=interests_two_months_free&utm_term=sticky_banner_launch)

[learn/master-cpp?](#)

[.com&utm_medium=referral&utm_audience=ORGANIC-sign=course_promotion&utm_content=interests_learn_cpp&utm_term=nav_floating_button](#))

denominator == 0

```
try {  
    if (denominator == 0)  
        throw 0;  
    // code  
}  
catch (int num_exception) {  
    // code  
}  
return 0;
```

denominator != 0

```
try {  
    if (denominator == 0)  
        throw 0;  
    // code  
}  
catch (int num_exception) {  
    // code  
}  
return 0;
```

Working of try, throw, and catch statements in C++

Notice that we have thrown the `int` literal **0** with the code `throw 0;`.

We can throw any literal or variable or class, depending on the situation and depending on what we want to execute inside the `catch` block.

The `catch` parameter `int num_exception` takes the value passed by the `throw` statement i.e. the literal **0**.

Thank you for printing our content at www.domain-name.com. Please check back soon for new contents.



3d: 19h: 20m: 21s

Stop copy pasting code you don't actually understand

Build the coding confidence you need to become a developer companies will fight
for

Become a PRO

(https://programiz.pro/offer/january-2026?utm_source=programiz.com&utm_medium=referral&utm_audience=ORGANIC-FREEMIUM&utm_campaign=newyear-26&utm_content=interests_two_months_free&utm_term=sticky_banner_launch)

[learn/master-cpp?](#)

[.com&utm_medium=referral&utm_audience=ORGANIC-align=course_promotion&utm_content=interests_learn_cpp&utm_term=nav_floating_button](#))

can occur due to the code in our `try` statement.

This is so that we can use the appropriate `catch` parameters. Otherwise, the `try...catch` statements might not work properly.

If we do not know the types of exceptions that can occur in our `try` block, then we can use the ellipsis symbol `...` as our `catch` parameter.

```
try {
    // code
}
catch (...) {
    // code
}
```

C++ Multiple catch Statements

In C++, we can use multiple `catch` statements for different kinds of exceptions that can result from a single block of code.

Thank you for printing our content at www.domain-name.com. Please check back soon for new contents.



3d: 19h: 20m: 21s

Stop copy pasting code you don't actually understand

Build the coding confidence you need to become a developer companies will fight
for

Become a PRO

(https://programiz.pro/offer/january-2026?utm_source=programiz.com&utm_medium=referral&utm_audience=ORGANIC-FREEMIUM&utm_campaign=newyear-26&utm_content=interests_two_months_free&utm_term=sticky_banner_launch)

[learn/master-cpp?](#)

[.com&utm_medium=referral&utm_audience=ORGANIC-align=course_promotion&utm_content=interests_learn_cpp&utm_term=nav_floating_button](#))

of `catch (...) {}` is executed.

Notes:

- `catch (...) {}` should always be the final block in our `try...catch` statement. This is because this block catches all possible exceptions and acts as the default `catch` block.
- It is not compulsory to include the default `catch` block in our code.

Example 2: C++ Multiple catch Statements

This program divides two numbers and stores the result in an array element.

There are two possible exceptions that can occur in this program:

- If the array is out of bounds i.e. if the index of the array is greater than the size of the array
- If a number is divided by 0

These exceptions are caught in multiple `catch` statements.

Thank you for printing our content at www.domain-name.com. Please check back soon for new contents.



3d: 19h: 20m: 21s

Stop copy pasting code you don't actually understand

Build the coding confidence you need to become a developer companies will fight
for

Become a PRO

(https://programiz.pro/offer/january-2026?utm_source=programiz.com&utm_medium=referral&utm_audience=ORGANIC-FREEMIUM&utm_campaign=newyear-26&utm_content=interests_two_months_free&utm_term=sticky_banner_launch)

[learn/master-cpp?](#)

[.com&utm_medium=referral&utm_audience=ORGANIC-align=course_promotion&utm_content=interests_learn_cpp&utm_term=nav_floating_button](#))

```
cout << "Enter numerator: ";
cin >> numerator;

cout << "Enter denominator: ";
cin >> denominator;

// throw exception if denominator is 0
if (denominator == 0)
    throw 0;
```

[Run Code »](#) ([/cpp-programming/online-compiler](#))

Output 1

```
Enter array index: 5
Error: Array out of bounds!
```

Here, the array `arr` only has **4** elements. So, `index` cannot be greater than **3**.

In this case, `index` is **5**. So we throw a string literal `"Error: Array out of bounds!"`. This exception is caught by the first `catch` block.

Notice the `catch` parameter `const char* msg`. This indicates that the `catch` statement takes a `string` literal as an argument.

Output 2

Thank you for printing our content at www.domain-name.com. Please check back soon for new contents.



3d: 19h: 20m: 21s

Stop copy pasting code you don't actually understand

Build the coding confidence you need to become a developer companies will fight
for

Become a PRO

(https://programiz.pro/offer/january-2026?utm_source=programiz.com&utm_medium=referral&utm_audience=ORGANIC-FREEMIUM&utm_campaign=newyear-26&utm_content=interests_two_months_free&utm_term=sticky_banner_launch)

[learn/master-cpp?](#)

[.com&utm_medium=referral&utm_audience=ORGANIC-align=course_promotion&utm_content=interests_learn_cpp&utm_term=nav_floating_button](#))

Here, the program runs without any problem as no exception occurs.

C++ Standard Exception

C++ has provided us with a number of standard exceptions that we can use in our exception handling. Some of them are shown in the table below.

Exception	Description
<code>std::exception</code>	The parent class of all C++ exceptions.
<code>std::bad_alloc</code>	Thrown when a dynamic memory allocation fails.
<code>std::bad_cast</code>	Thrown by C++ when an attempt is made to perform a <code>dynamic_cast</code> to an invalid type.
<code>std::bad_exception</code>	Typically thrown when an exception is thrown and it cannot be rethrown.

There are many other standard exceptions in C++.

These exceptions are defined in the `exception` header file.

Thank you for printing our content at www.domain-name.com. Please check back soon for new contents.



3d: 19h: 20m: 21s

Stop copy pasting code you don't actually understand

Build the coding confidence you need to become a developer companies will fight
for

Become a PRO

(https://programiz.pro/offer/january-2026?utm_source=programiz.com&utm_medium=referral&utm_audience=ORGANIC-FREEMIUM&utm_campaign=newyear-26&utm_content=interests_two_months_free&utm_term=sticky_banner_launch)

[learn/master-cpp?](#)

[.com&utm_medium=referral&utm_audience=ORGANIC-align=course_promotion&utm_content=interests_learn_cpp&utm_term=nav_floating_button](#))

[Previous](#)

[Tutorial:](#)

[C++](#) [\(/cpp-programming/functors\)](#)

[Functor](#)

[Next Tutorial:](#)

[C++ File](#)

[Handling](#)

[\(/cpp-programming/file-handling\)](#)

Share on:

(<https://www.facebook.com/sharer/sharer.php?u=https://www.programiz.com/cpp-programming/exception-handling>)

(<https://twitter.com/intent/tweet?url=https://www.programiz.com/cpp-programming/exception-handling&text=Check%20this%20amazing%20programming/exception-handling>)

Did you find this article helpful?



Programiz PRO

Your builder path starts here.

Escape tutorial hell and ship real projects.

Try Programiz PRO

Thank you for printing our content at www.domain-name.com. Please check back soon for new contents.



3d: 19h: 20m: 21s

Stop copy pasting code you don't actually understand

Build the coding confidence you need to become a developer companies will fight
for

Become a PRO

(https://programiz.pro/offer/january-2026?utm_source=programiz.com&utm_medium=referral&utm_audience=ORGANIC-FREEMIUM&utm_campaign=newyear-26&utm_content=interests_two_months_free&utm_term=sticky_banner_launch)

[learn/master-cpp?](#)

[.com&utm_medium=referral&utm_audience=ORGANIC-align=course_promotion&utm_content=interests_learn_cpp&utm_term=nav_floating_button](#))

([/cpp-programming/if-else](#))

[C++ Tutorial](#)

C++ std::array

([/cpp-programming/namespaces](#))

[C++ Tutorial](#)

C++ Vectors

([/cpp-programming/std-array](#))

([/cpp-programming/vectors](#))

