

# C++ Exception Handling

Exception Handling in C++ is a process to handle runtime errors. We perform exception handling so the normal flow of the application can be maintained even after runtime errors.

In C++, exception is an event or object which is thrown at runtime. All exceptions are derived from `std::exception` class. It is a runtime error which can be handled. If we don't handle the exception, it prints exception message and terminates the program.

---

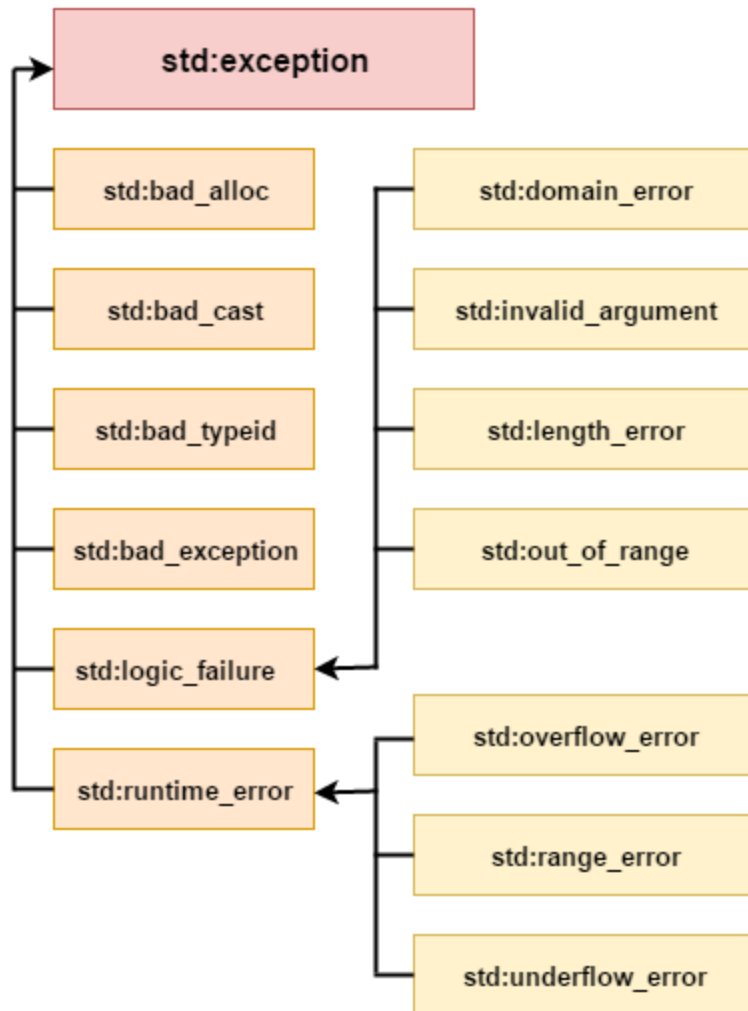
## Advantage

It maintains the normal flow of the application. In such case, rest of the code is executed even after exception.

---

## C++ Exception Classes

In C++ standard exceptions are defined in `<exception>` class that we can use inside our programs. The arrangement of parent-child class hierarchy is shown below:



All the exception classes in C++ are derived from `std::exception` class. Let's see the list of C++ common exception classes.

| Exception                       | Description   |
|---------------------------------|---|
| <code>std::exception</code>     | It is an exception and parent class of all standard C++ exceptions. |
| <code>std::logic_failure</code> | It is an exception that can be detected by reading a code.          |
| <code>std::runtime_error</code> | It is an exception that cannot be detected by reading a code.       |
| <code>std::bad_exception</code> | It is used to handle the unexpected exceptions in a c++ program.    |
| <code>std::bad_cast</code>      | This exception is generally be thrown by <b>dynamic_cast</b> .      |

|                              |  |
|------------------------------|--|
| <code>std::bad_typeid</code> | This exception is generally be thrown by <b>typeid</b> . |
| <code>std::bad_alloc</code>  | This exception is generally be thrown by <b>new</b> .    |

---

## C++ Exception Handling Keywords

In C++, we use 3 keywords to perform exception handling:

- try
- catch, and
- throw

## C++ try/catch

In C++ programming, exception handling is performed using try/catch statement. The C++ **try block** is used to place the code that may occur exception. The **catch block** is used to handle the exception.

---

## C++ example without try/catch

```
1. #include <iostream>
2. using namespace std;
3. float division(int x, int y) {
4.     return (x/y);
5. }
6. int main () {
7.     int i = 50;
8.     int j = 0;
9.     float k = 0;
10.    k = division(i, j);
11.    cout << k << endl;
12.    return 0;
13. }
```

Output:

```
Floating point exception (core dumped)
```

---

## C++ try/catch example

```
1. #include <iostream>
2. using namespace std;
3. float division(int x, int y) {
4.     if( y == 0 ) {
5.         throw "Attempted to divide by zero!";
6.     }
7.     return (x/y);
8. }
9. int main () {
10.    int i = 25;
11.    int j = 0;
12.    float k = 0;
13.    try {
14.        k = division(i, j);
15.        cout << k << endl;
16.    }catch (const char* e) {
17.        cerr << e << endl;
18.    }
19.    return 0;
20.}
```

Output:

```
Attempted to divide by zero!
```