

Exception in C++

When executing C++ code, different errors can occur: coding errors made by the programmer, errors due to wrong input, or other unforeseeable things.

When an error occurs, C++ will normally stop and generate an error message. The technical term for this is: C++ will throw an exception (**throw an error**).

C++ supports three keywords for this purpose : “try”, “throw”, “catch”

The **try** statement allows you to define a block of code to be tested for errors while it is being executed.

The **throw** keyword throws an exception when a problem is detected, which lets us create a custom error.

The **catch** statement allows you to define a block of code to be executed, if an error occurs in the try block.

NOTE: A function-try-block associates a sequence of catch clauses with the entire function body, and with the member initializer list (if used in a constructor) as well. Every exception thrown from any statement in the function body, or (for constructors) from any member or base constructor, or (for destructors) from any member or base destructor, transfers control to the handler-sequence the same way an exception thrown in a regular try block would. (**TRY and CATCH must be in same function / block otherwise error generated**)

Example:

```
try {  
    int age = 15;  
    if (age >= 18) {  
        cout << "Access granted - you are old enough.";  
    } else {  
        throw 505;  
    }  
}  
  
catch (int myNum) {  
    cout << "Access denied - You must be at least 18 years old.\n";  
    cout << "Error number: " << myNum;  
}
```

When you want to handle all error with single block:

```
try {  
    int age = 15;  
  
    if (age >= 18) {  
  
        cout << "Access granted - you are old enough.";  
  
    } else {  
  
        throw 505;  
  
    }  
  
    catch (...) {  
  
        cout << "Access denied - You must be at least 18 years old.\n";  
  
    }  
}
```