**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.

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

* try
* catch, and
* throw

**1. try**

The try block contains the code that might throw an exception. If an exception occurs, the control is transferred to the catch block.

**2. throw**

The throw keyword is used to signal that an exception has occurred. It is followed by an argument that indicates the exception type or value.

**3. catch**

The catch block is used to handle the exception. It specifies the type of exception it can handle.

**4. ... (Ellipsis in catch)**

C++ provides the ... syntax to catch exceptions of any type.

**Example :**

**#include <iostream>**

**using namespace std;**

**int main() {**

**try {**

**// Code that may throw an exception**

**int a = 10, b = 0;**

**if (b == 0)**

**throw string("Division by zero error!");**

**cout << "Result: " << a / b << endl;**

**} catch (string msg) {**

**// Catch block handles the exception**

**cout << "Exception: " << msg << endl;**

**}**

**return 0;**

**}**

**o/p**

Exception: Division by zero error!

**Example 2: Catching Multiple Exceptions**

#include <iostream>

using namespace std;

int main() {

try {

int choice;

cout << "Enter 1 for int exception, 2 for string exception: ";

cin >> choice;

if (choice == 1)

throw 100; // Throwing an int

else if (choice == 2)

throw string("String exception!");

else

throw "Unknown exception!";

} catch (int e) {

cout << "Caught an integer: " << e << endl;

} catch (string e) {

cout << "Caught a string: " << e << endl;

} catch (...) {

cout << "Caught an unknown exception!" << endl;

}

return 0;

}

1. A try block must always be followed by one or more catch blocks.

2. Exceptions can be of any type (e.g., int, string, or custom objects).