C++ Try Catch

C++ Try Catch

C++ Try Catch statement is used as a means of exception handling.

You may come across some exceptional situations where you may not have control of the values for a variable or such. And this could result in anomalies that C++ cannot execute. In such conditions, C++ throws an exception, and could stop the execution of program.

If we are aware of such typical programming conditions, and if we have a tool to handle the exception, and move on with the execution, it would be really great. And C++ way of handling exceptions is try-catch.

Please note that Try Catch in C++ is quite different, in terms of inbuilt exceptions, from that of in programming languages like Java, Python, etc.

In this tutorial, we learn the syntax of try catch statement in C++, and how to use try catch statement to catch exceptions.

Syntax of C++ Try Catch

Following is the syntax of Try Catch statement.

```
try {
   // statement(s)
} catch (ExceptionName e) {
   // statement(s)
}
```

In try block, write statements that have potential to throw an exception during runtime. Or, you could throw an exception.

In the catch block, you can write statements to make your program take a course when this exception occurs. And continue with the rest of the program, after this try catch statement.

In addition, you can have multiple catch blocks, followed by try block, handling different type of exceptions. You may write multiple catch blocks, when your try code can throw different types of exceptions.

Following is the syntax of try catch statement with multiple catch blocks.

```
try {
   // statement(s)
} catch (ExceptionName1 e) {
   // statement(s)
} catch (ExceptionName2 e) {
   // statement(s)
} catch (ExceptionName3 e) {
   // statement(s)
}
```

Throw Exception

You can throw exception of any primitive datatype or an object of custom class type.

Following is the syntax of throw statement.

```
throw exception;
```

C++ Try Catch Example

In this example, we shall try dividing a number with another. Before executing division, we shall check if the denominator is zero and throw an exception if so.

C++ Program

```
#include <iostream>
using namespace std;

int main() {
   int a = 10;
   int b = 0;

   try {
      if (b == 0) {
         throw 0;
      }
      cout << a/b << endl;
   } catch (int n) {
      cout << "Denominator is zero. We cannot perform division." << endl;
   }
}
```

Output

```
Denominator is zero. We cannot perform division.
```

C++ Try Catch with Multiple Exceptions

In this example, we shall try dividing a number with another. Before executing division, we shall check if the denominator is zero. Throw an exception, if so, of `int` type. Also, we shall check if numerator is zero and throw an exception of `char const*` type.

The catch block that matches the thrown exception is executed.

C++ Program

```
#include <iostream>
using namespace std;

int main() {
   int a = 0;
   int b = 5;

   try {
      if (b == 0) {
         throw 0;
      }
      cout << a/b << endl;

      if (a/b==0) {
         throw "There is nothing to divide among.";
      }
   } catch (int n) {
      cout << "Denominator is zero. We cannot perform division." << endl;
   } catch (char const* s) {
      cout << s << endl;
   }
}
```

Output

```
0
There is nothing to divide among.
```

Conclusion

In this C++ Tutorial, we learned what a try catch statement is in C++, how does it handle exceptions, and how to throw and catch exceptions using try catch.