Exceptions Handling Definition:

In C++ Programming, Exception Handling is the best method for handles run-time errors.  It provides a method to convey control from try block of a program to catch block. C++ implements following three specific keywords for exception handling: try, catch, and throw.

try block:

* The try keyword is used to declare a block of statements, which code may produce exceptions.
* When an exception is identified, It is thrown using a throw statement in the try Block. program control immediately passes to catch block from the try block.

a try block syntax looks like the following:

try {

*//main code*

}

Catch block:

The catch block represented by the keyword catch. Catch block catches exception the exception thrown by the throw statement in the try block and catch block handles the exception.

a catch block syntax looks like the following:

try {

} catch (ExceptionType1 name) {

*//exception code*

} catch (ExceptionType2 name) {

*//exception code*

}

throw Statement:

Possible to declare throw keyword anywhere in the try block. throw keyword trigger/execute a matched type of the catch block.The try block exists When a throw statement is reached.

a throw statement syntax looks like the following:

try {

throw data\_value;

} catch (ExceptionType1 name) {

*//exception code*

} catch (ExceptionType2 name) {

*//exception code*

}

Exception Handling Checklist

* Try block has the main code, which code may throw an exception.
* Multiple throws are possible to declare in try block based different situations.
* The catch block contains exception handling code that is executed when thrown by the Try Block.
* Possible to declare one or more catch block for different type of exception
* No code can be between try block and the first catch block.
* Catch blocks declare directly after the try block.

Simple Exception Program In C++

*// Simple Exception Program In C++*

**#include <iostream>**

using namespace std;

int **main**() {

*//try block*

try {

*// Exception throwed*

throw 100;

} *//catch block*

catch (int exception\_value) {

cout << "Exception Occurred : Exception Value : " << exception\_value;

}

return 0;

}

Exception Occurred : Exception Value : 100

Multiple Catch Block Exception Program In C++

*// Multiple Catch Exception Example Program (Divide By Zero ) In C++*

**#include <iostream>**

using namespace std;

float **divide**(int x, int y) {

if (y == 0) {

throw y;

} else if (y < 0) {

throw "Negative Input";

}

return (x / y);

}

int **main**() {

int i, result;

*//try block*

cout << "Enter the Number :";

cin>>i;

try {

result = divide(100, i);

cout << result << endl;

} *//catch block*

catch (int exception\_value) {

cout << "Exception Occurred : Exception Value : " << exception\_value;

} catch (const char\* excpection\_str) {

cout << "Exception Occurred : Exception Value : " << excpection\_str;

}

return 0;

}

*/\**

*Execution Scenarios*

*// Scenario 1*

*result = divide(100, 10);*

*cout << result << endl;*

*// Scenario 2*

*result = divide(100, 0);*

*cout << result << endl;*

*// Scenario 3*

*result = divide(100, -1);*

*cout << result << endl;*

*\*/*

Sample Output:

Scenario 1:

Enter the Number :10

10

Scenario 2:

Enter the Number :0

Exception Occurred : Exception Value: 0

Scenario 3:

Enter the Number :-1

Exception Occurred : Exception Value: Negative Input