Week 10: Exception Handling

Part 1. Exception Handling
An exception is a problem that arises during the execution of a program. A C++ exception is a response to an exceptional circumstance that arises while a program is running, such as an attempt to divide by zero. Exceptions provide a way to transfer control from one part of a program to another. C++ exception handling is built upon three keywords: try, catch, and throw.

- **throw**: A program throws an exception when a problem shows up. This is done using a throw keyword.
- **catch**: A program catches an exception with an exception handler at the place in a program where you want to handle the problem. The catch keyword indicates the catching of an exception.
- **try**: A try block identifies a block of code for which particular exceptions will be activated. It's followed by one or more catch blocks.

Assuming a block will raise an exception, a method catches an exception using a combination of the try and catch keywords. A try/catch block is placed around the code that might generate an exception. Code within a try/catch block is referred to as protected code, and the syntax for using try/catch looks like the following:

```cpp
try
{
    // protected code
} catch( ExceptionName e1 )
{
    // catch block
} catch( ExceptionName e2 )
{
    // catch block
} catch( ExceptionName eN )
{
    // catch block
}
```
You can list down multiple `catch` statements to catch different type of exceptions in case your `try` block raises more than one exception in different situations.

- **Catching Exceptions**

The `catch` block following the `try` block catches any exception. You can specify what type of exception you want to catch and this is determined by the exception declaration that appears in parentheses following the keyword `catch`.

```cpp
try {
    // protected code
} catch (ExceptionName e) {
    // code to handle ExceptionName exception
}
```

Above code will catch an exception of `ExceptionName` type. If you want to specify that a catch block should handle any type of exception that is thrown in a try block, you must put an ellipsis, `...`, between the parentheses enclosing the exception declaration as follows:

```cpp
try {
    // protected code
} catch (...) {
    // code to handle any exception
}
```

The following is an example, which throws a division by zero exception and we catch it in catch block.

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

double division(int a, int b) {
    if (b == 0) {
        throw "Division by zero condition!";
    }
    return (a/b);
}

int main() {
    int x = 50;
    int y = 0;
    double z = 0;
    try {
        z = division(x, y);
        cout << z << endl;
    } catch (const char* msg) {
        cerr << msg << endl;
    }
    return 0;
}
```
Because we are raising an exception of type `const char*`, so while catching this exception, we have to use `const char*` in catch block. If we compile and run above code, this would produce the following result:

```
Division by zero condition!
```