Object-Oriented Programming Using C++

Exception Handling

Indranil Saha

Department of Computer Science and Engineering
Indian Institute of Technology Kanpur

Exception

- 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

Exception Handling in C++

- Exceptions provide a way to transfer control from one part of a program to another.
- C++ exception handling is built upon three keywords
  - **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.

Throwing Exception

Output

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

Catching Exception

Handling a Specific Exception

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

Handling Any Exception

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

Example: Exception Handling in C++

Division-by-zero exception handling

```
#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;
}
```

C++ Standard Exceptions

- **std::exception**
  - An exception and parent class of all the standard C++ exceptions
- **std::bad_alloc**
  - This can be thrown by new.
- **std::domain_error**
  - This is an exception thrown when a mathematically invalid domain is used
- **std::invalid_argument**
  - This is thrown due to invalid arguments
- **std::overflow_error**
  - This is thrown if a mathematical overflow occurs
- **std::range_error**
  - This is occurred when you try to store a value which is out of range

Object-Oriented Programming Using C++

Exception Handling

Indranil Saha

Department of Computer Science and Engineering
Indian Institute of Technology Kanpur