# Exceptions in C++

## Basic Exception Handling

- C++ supports exceptions in a manner similar to many other languages

```
try
{
  result = ACriticalFunction();
  if ( result != AGoodValue )
     throw( "Bad result" );
  NextBitOfCode();
}
catch( const char * pszMessage )
{
   cout << "Problem, " << pszMessage << endl;
}
```

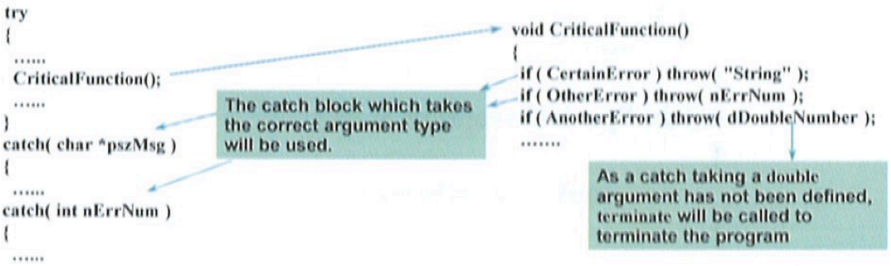
A 'try' block is placed around code where an exception could occur

If the application detects an exception condition, it calls 'throw', passing an argument. The program jumps to the following 'catch' block, leaving the 'try' block in a clean state (local variables are correctly destructed, for example)

The 'catch' block handles the exception in an appropriate way.

## The Traditional Approach

- 'catch' takes an argument that can be of any type
  - catch(…) will catch any thrown exception
- 'throw' takes an argument of any type
- Exception is caught by nearest enclosing catch for the correct type
  - If no match found, program is terminated




## Exceptions in the Standard Library

- Standard library defines class named exception
  - Standard exceptions are defined as types deriving from this
  - Additional exception types can be defined
- Member function what() returns description of exception oas a char *
  - Application defined exceptions should override this

```
class SpecialException : public exception
{
public:
        const char * what()
        {
                return "Oops!";
        }
};
```

## Throwing and Catching the Exception

```
try
{
      if ( itWentWrong )
      {
         SpecialException se;
         throw se;
      }
         ...
}
catch( SpecialException & ex )
{
          cout << ex.what();
}
```

## Standard Exception Classes

- Deriving from std::exception
- logic_error
  - Base class for domain_error, invalid_argument, length_error, out_of_range, future_error
- runtime_error
  - Base class for overflow_error, underflow_error, range_error, system_error
  - ios_base::failure derived from system_error
- bad_alloc, bad_cast, bad_exception, bad_function_call, bad_weak_ptr, bad_typeid
  - bad_array_new_length derived from bad_alloc