Lecture 11.23

Exception Handling

SE271 Object-Oriented Programming (2020)

Yeseong Kim

Original slides from Prof. Shin at DGIST

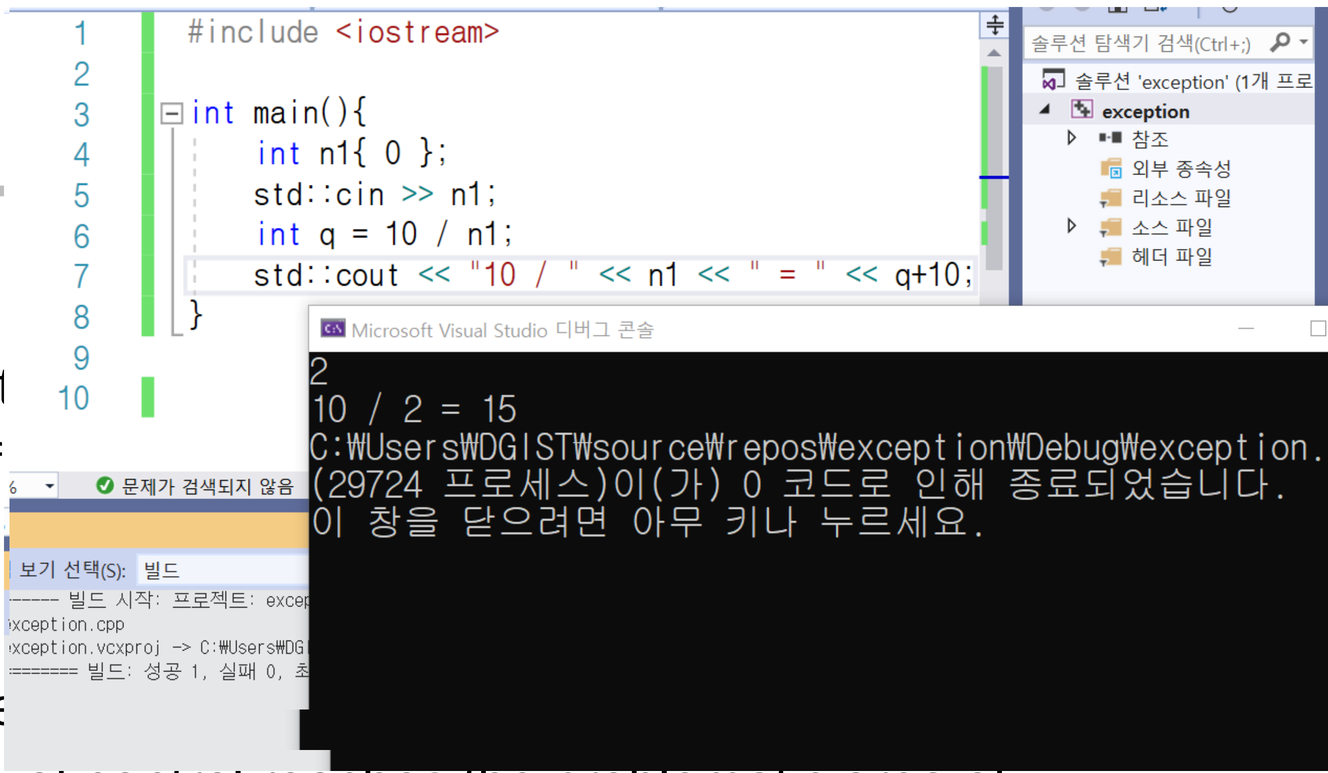
Short Notice

- Team Project – Guideline will be released by this Friday, **but please prepare in advance!**
  - Will have a presentation with a recorded video (4 minutes for each team)
  - Will write a report (3~5 pages)

Today’s Topic

- Exception Handling
  - Exception Class

[recap] Types of Errors



```
#include <iostream>

int main(){
    int n1{ 0 };
    std::cin >> n1;
    int q = 10 / n1;
    std::cout << "10 / " << n1 << " = " << q+10;
}
```

```
------ 빌드 시작: 프로젝트: excep
xception.cpp
xception.vcxproj -> C:₩Users₩DG
======= 빌드: 성공 1, 실패 0, 초
```

```
2
10 / 2 = 15
C:₩Users₩DGIST₩source₩repos₩exception₩Debug₩exception.
(29724 프로세스)이(가) 0 코드로 인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

- Syntax errors (or syntactic errors)
  - If a program does not follow the synt
  - Detected by compiler/interpreter bef
  - Easiest to find and fix
- Runtime errors **How to avoid?**
  - Errors occur as the program execute
  - Errors are not detected until the flow of control reaches the problematic area of codes
  - Typically errors occurs when a certain condition holds (e.g., divide by zero)
- Semantic errors (or logical errors)
  - Usually program runs without any error, but it does not produce correct answers
  - Caused by a wrong design or implementation of the program
  - Most difficult to find and fix

```
#include <iostream>
#include <fstream>
#include <string>
#include <iomanip>
using namespace std;
int main() {
    string name, pn;
    int age;
    ofstream fout(“Phonebook.bin", ios::app|ios::binary);
    getline(cin, name);
    cin >> age;
    cin.ignore(100, '\n');
    getline(cin, pn);

    fout << setw(10) << left << name;
    fout << setw(3) << age;
    fout << setw(13) << pn;
}
```

```
Alice Kim
20
010-111-2222
Bob Lee
21
010-1111-2222
```

```
Alice Kim 20 010-111-2222 Bob Lee   21 010-
1111-2222
```

What if Name includes digit?

What if age is set by negative value?

Non-number

What if phone number is strange?

More than 13 words

Input Validation

```
int main() {
    string name, pn;
    int age;
    ofstream fout(“Phonebook.bin",
ios::app|ios::binary);
    getline(cin, name);
    cin >> age;
    cin.ignore(100, '\n');
    getline(cin, pn);

    fout << setw(10) << left << name;
    fout << setw(3) << age;
    fout << setw(13) << pn;
}
```

```
If (name is valid?) {
    Processing…Ask age?

    if (age is valid?) {
        Processing…Ask P/N?

        if (phone number is valid?) {

        } else …

    } else …

} else …
```

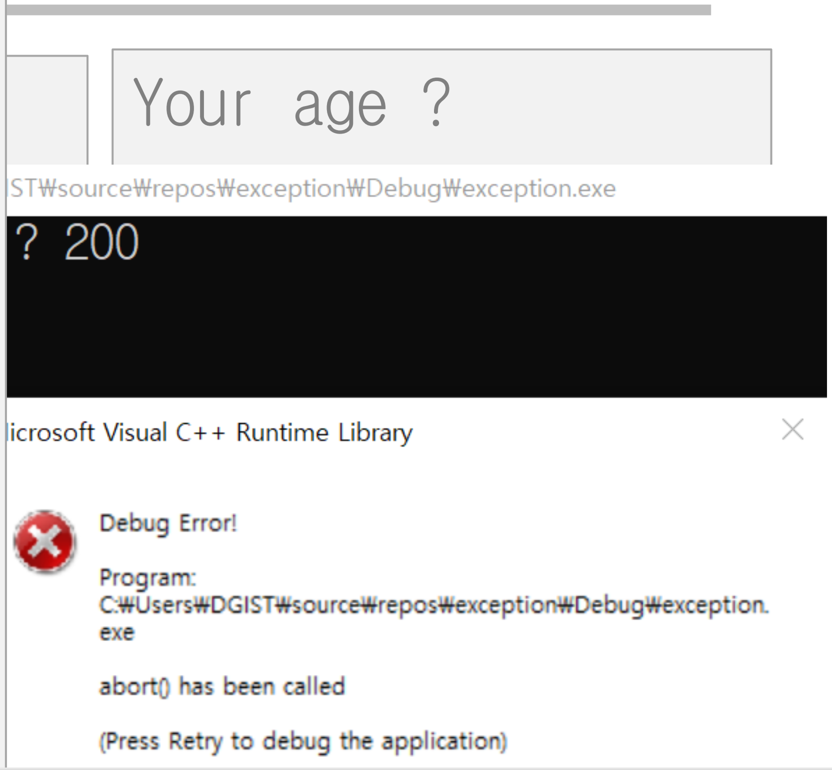
Exception Handling

- Syntax

```
try
{
    if there is an error, a system throws exceptions…
    or a program throws exceptions…
}
catch (exception_parameter)
{


}
…
```

Example: Exception

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

int main() {
    int age;
    cout << "Your age ? ";
    try{
        cin >> age;
        if (age < 0 || age >= 150)
            throw age;
        cout << "Age: " << age;
    }
    catch(int i) {
        cout << "Age( " << age << " ) is not valid. ";
    }
}
```



```
Your age ?
```

```
ST₩source₩repos₩exception₩Debug₩exception.exe
? 200
```

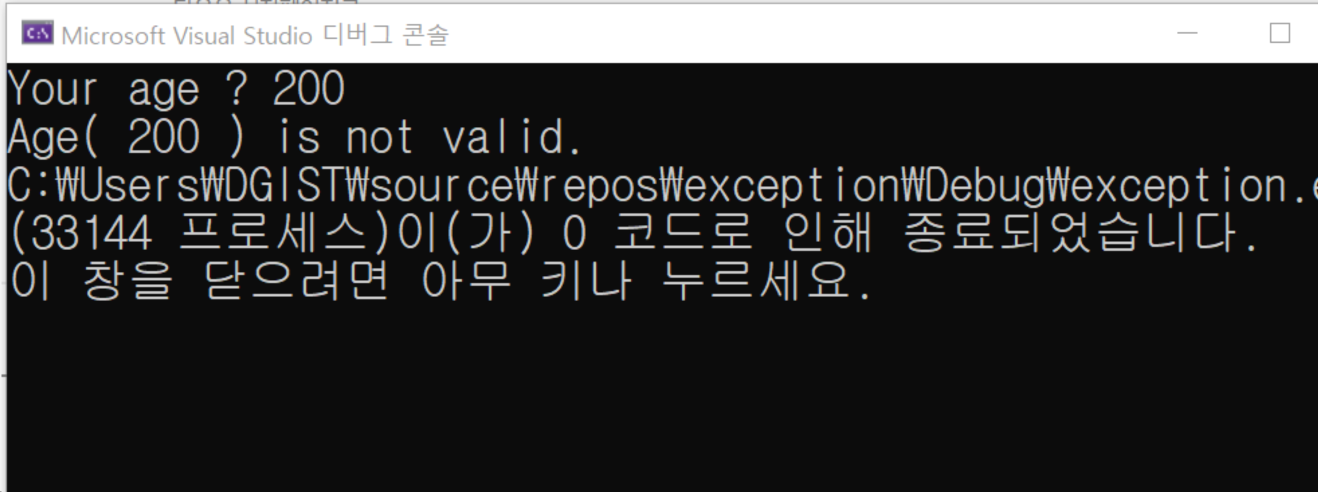
licrosoft Visual C++ Runtime Library

Debug Error!

Program:
C:₩Users₩DGIST₩source₩repos₩exception₩Debug₩exception.
exe

abort() has been called

(Press Retry to debug the application)



```
Your age ? 200
Age( 200 ) is not valid.
C:₩Users₩DGIST₩source₩repos₩exception₩Debug₩exception.e
(33144 프로세스)이(가) 0 코드로 인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

Multiple catch

- Syntax

```
try {
    if there is an error, a system throws exceptions…
    or a program throws exceptions…
}
catch (exception_parameter) {
}
catch (exception_parameter) {
}
catch (…) {
}
…
```

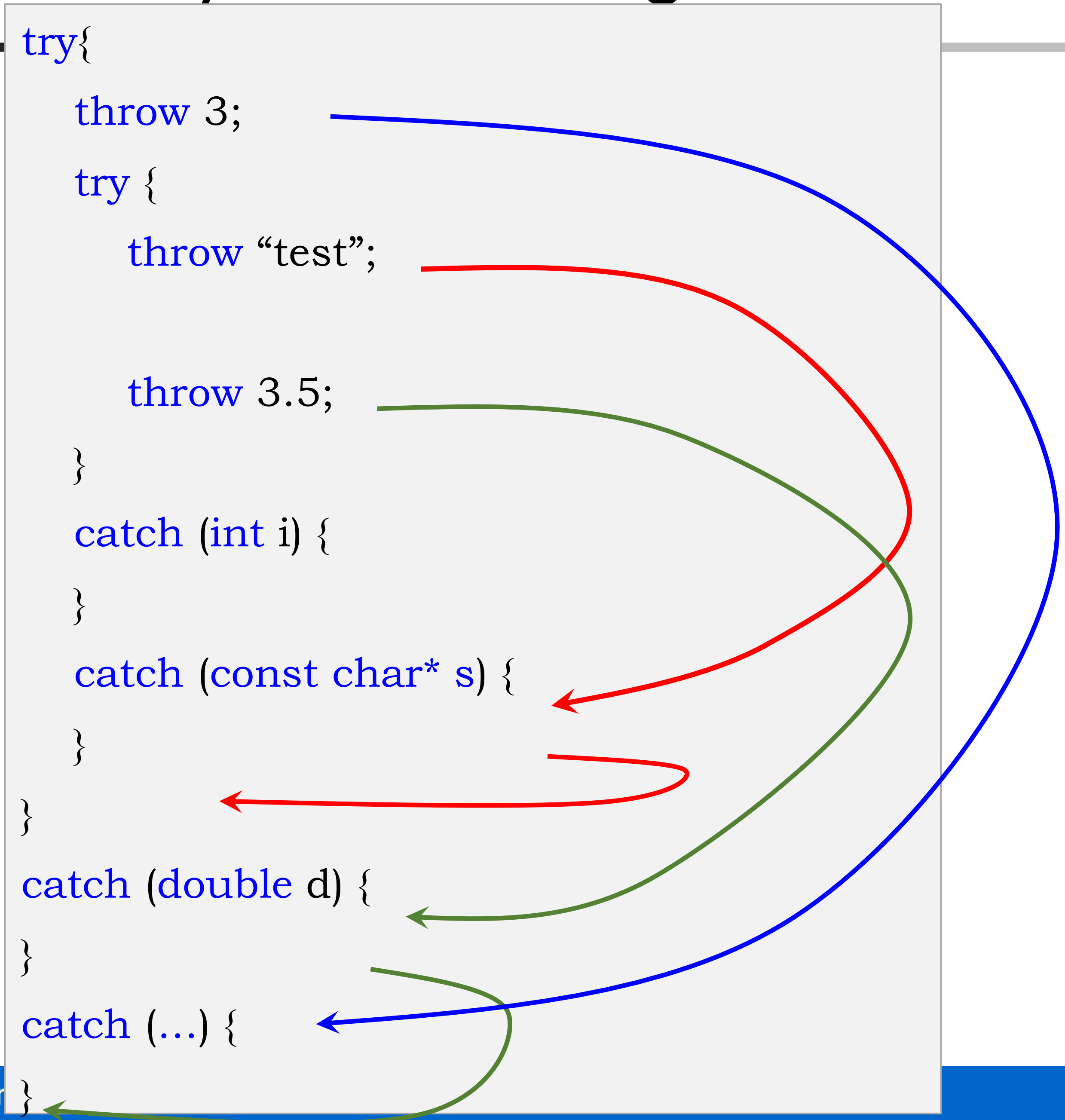
Catch-all handlers

Example: multiple catches

```
int main() {
    string name;
    int age;
    try{
        cout << "Your name ? ";
        getline(cin, name);
        if (name.length() > 10) throw name;
        cout << "Your age ? ";
        cin >> age;
        if (age < 0 || age >= 150)throw age;
        cout << "Age: " << age;
    }
    catch (int i) {
       cout << "Age( " << i <<  " ) is not valid. ";
    }
}
```

```
int main() {
    string name;
    int age;
    try{
        cout << "Your name ? ";
        getline(cin, name);
        if (name.length() > 10) throw name;
        cout << "Your age ? ";
        cin >> age;
        if (age < 0 || age >= 150)throw age;
        cout << "Age: " << age;
    }
    catch (int i) {
       cout << "Age( " << i <<  " ) is not valid. ";
    }
    catch (string s) {
       cout << “Name( " << s <<  " ) is not valid. ";
    }
}
```

Nested Try and Catching



```
try{
    throw 3;
    try {
        throw “test”;

        throw 3.5;
    }
    catch (int i) {
    }
    catch (const char* s) {
    }
}
catch (double d) {
}
catch (...) {
}
```

stack unwinding

```
void func01() {
    throw “func01";
}
int main(){
    try{
        func01();
    }
    catch (const char* ex) {
        cout  << "Exception at " << e
    }
}
```

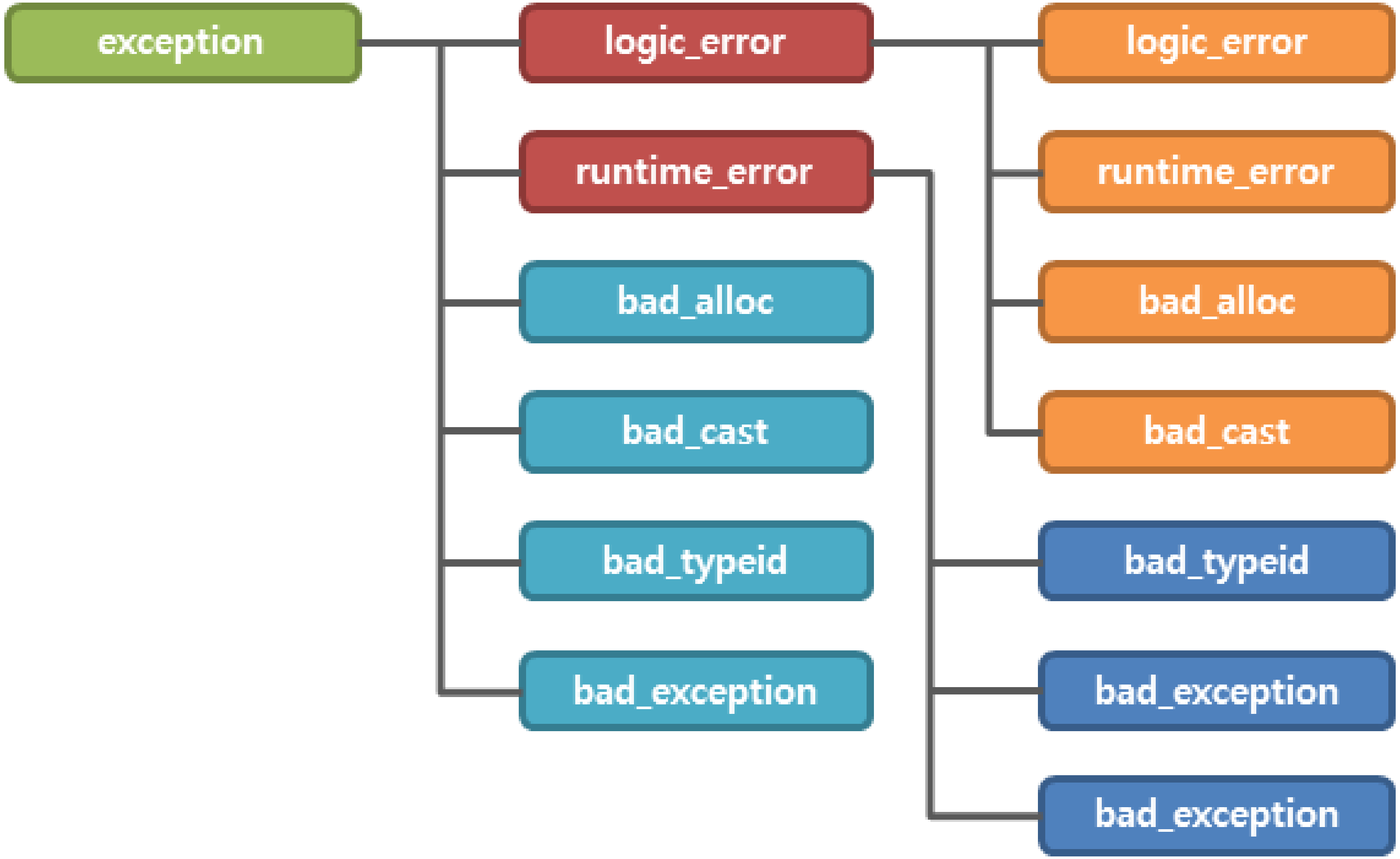
```
void func03() {
    throw “func03";
}
void func02() { func03(); }
void func01() { func02(); }
int main(){
    try{
        func01();
    }
    catch (const char* ex) {
        cout  << "Exception at " << ex  << endl;
    }
}
```

MyException Class

```
class MyException {
    int errNo;
    string errFunc, errMsg;
public:
    MyException(int n, string f, string m):
        errNo{ n }, errFunc{ f }, errMsg{ m }{}
    virtual ~MyException() {}
    void what(){
        cout << "Error[" << errNo << "] : "
<< errMsg << " at " << errFunc << endl;
    }
};
class MyDivideByZero : public MyException{
public:
    MyDivideByZero(string f) :
    MyException(100, f, "Divide by Zero") {} };
```

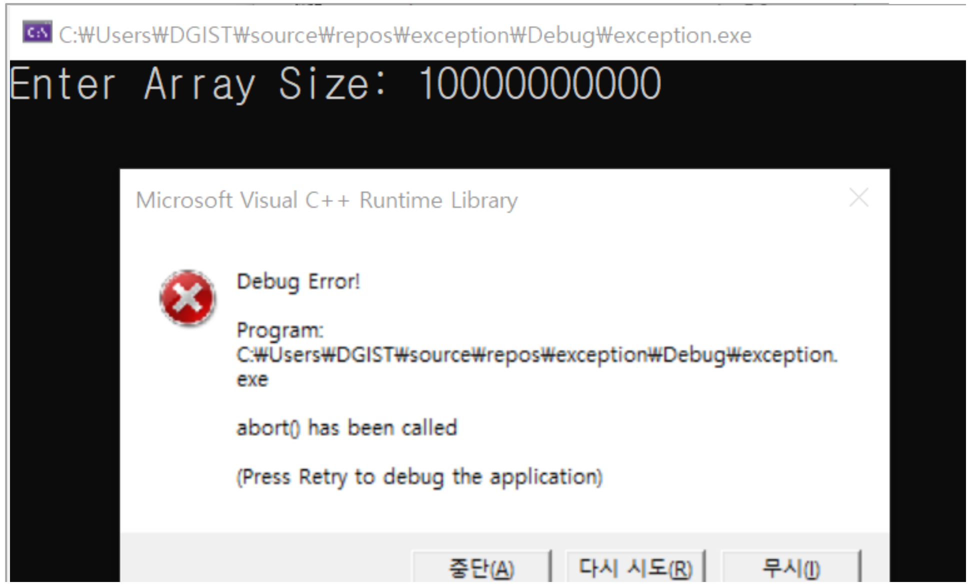
```
int main() {
    int n1{ 10 }, n2{ 0 };
    cin >> n2;

    try {
        if (n2 == 0)
            throw MyException(100, "main()", "Zero");
    }              MyDivideByZero("main()");
    catch (MyException & e) {
        e.what();
    }
}
```

Exception Class



Example: Exception Class

```
#include<exception>
int main() {
    int nSize;
    char* arr;
cout << "Enter Array Size: ";
    cin >> nSize;
    try {
        arr = new char[nSize];
        cout << "Array (" << _msize(arr) << ") is
created.";
    delete[] arr;
    }
    catch (bad_alloc & e) {
        cout << e.what() << endl;
    }
}
```



C:₩Users₩DGIST₩source₩repos₩exception₩Debug₩exception.exe

```
Enter Array Size: 10000000000
```

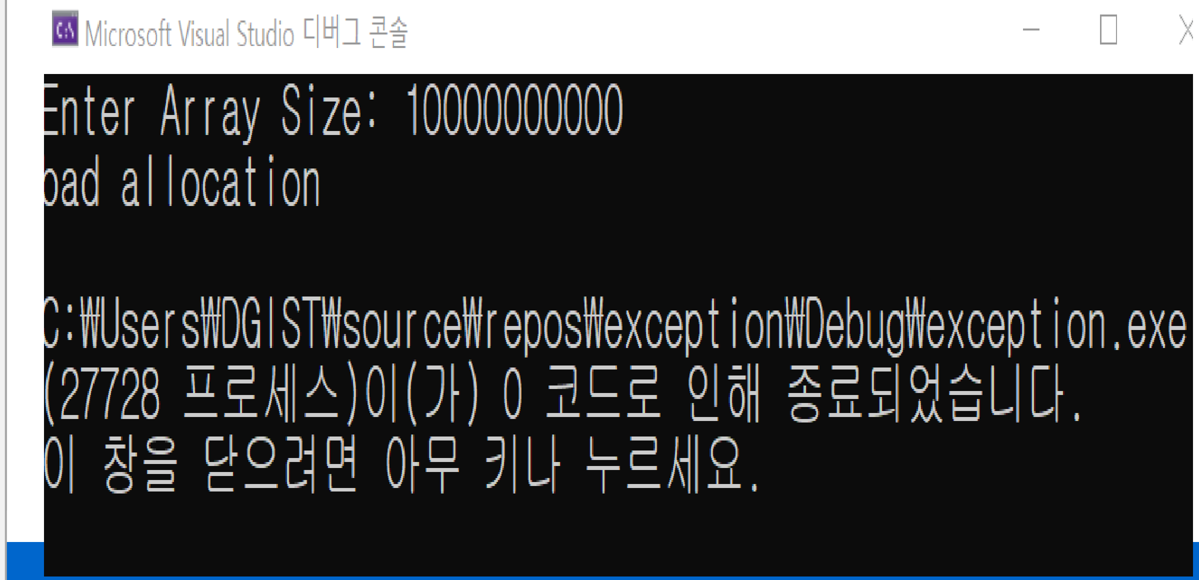
Microsoft Visual C++ Runtime Library

Debug Error!

Program:
C:₩Users₩DGIST₩source₩repos₩exception₩Debug₩exception.exe

abort() has been called

(Press Retry to debug the application)



Microsoft Visual Studio 디버그 콘솔

```
Enter Array Size: 10000000000
bad allocation

C:₩Users₩DGIST₩source₩repos₩exception₩Debug₩exception.exe(27728 프로세스)이(가) 0 코드로 인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

Inherit from Exception Class & noexcept

```
class  exception {
public:
    exception() noexcept;
    exception(const exception &) noexcept;
    virtual ~exception();
    virtual const char * what() const noexcept;
};
```

```
class MyException : public exception {
public:
    virtual const char* what() const noexcept {
        return “my exception”;
    }
};


try {
    throw MyException{};
}
catch (exception & e) {
    cout << e.what();
}
```

Uncatched Exception

```
#include <iostream>
#include <exception>
void MyErrorHandler() {
    std::cout << "Error anyway";
    exit(-1);
}
int main() {
    set_terminate(MyErrorHandler);
    try {
        throw 3.14;
    }
    catch (int i) {
    }
}
```

References

- Learn c++
  - https://www.learncpp.com/
  - Chapter : 14

Q

ANY QUESTIONS?