Exception handling

Nguyen Van Vu
nvu@fit.hcmus.edu.vn

Topics

- Try/catch/throw exceptions
- RAII
- Define new exceptions

Exceptions

- Is there any problem with the following code?

```
class Student {
    private:
        int m_Age;
    public:
        Student(int age) {
            m_Age = age;
        }
};
void main() {
    Student std(-10);

    char studentName [20] = ”Nguyen Minh”;
    cout << studentName [20];
}
```

Problems

1. A student has a negative age.
2. There is an out of range error.

Try/catch/throw exceptions

- Using try, catch, throw to handle exceptions in C++

```
try {
    // code may have errors/exceptions
} catch( ExceptionName exceptionType1 ) {
    // catch block
} catch( ExceptionName exceptionType2 ) {
    // catch block
} catch( ExceptionName exceptionType3 ) {
    // catch block
}
…
throw ExceptionName();
```

Try/catch/throw exceptions

- Using try, catch, throw to handle exceptions in C++

```
class Student {
    private:
            int m_Age;
    public:
            Student(int age) {
                        if (age <= 0) throw "Invalid age!";
                        m_Age = age;
            }
};
void main() {
    try {
            Student std(-10);

            char studentName [20] = "Nguyen Minh";
            cout << studentName [20];
    } catch (const char* error) { }
}
```

Standard exceptions

- Superclass: std::exception

logic_error	Member functions
invalid_argument	**constructor**
domain_error	**destructor**
length_error	**opertator =**
out_of_range	**what**

runtime_error
- **range_error**
- **overflow_error**
- **underflow_error**

bad_typeid
bad_cast
bad_alloc
bad_exception

Try/catch/throw exceptions

- Using try, catch, throw to handle exceptions in C++

```
#include <iostream>
#include <exception>
using namespace std;
class MyException : public exception {
    public:
    const char * what () const throw () {
            return ”My Exception Exception";
    }
};
```

```
int main() {
    try {
            throw MyException();
    } catch(MyException& e) {
            std::cout << "MyException";
            std::cout << e.what() << std::endl;

    } catch(std::exception& e) {
            // Handle other error
    }
}
```

Resource Acquisition Is Initialization (RAII)

```
int write_file () {
    std::FILE* file_handle = std::fopen(”text-file.txt", "w+");

    if (file_handle == NULL ) {
        throw “Open file error”;
    }
    try {
        if (std::fputs(”Write something", file_handle) == EOF ) {
            throw “Error when writing to file” ;
        }
    } catch(...) {
        std::fclose(file_handle); // close file
        throw; // re-throw exception
    }
    std::fclose(file_handle); // close file
}
```

1. What if there is an exception here?

2. We have to handle closing_file handle manually

Resource Acquisition Is Initialization (RAII)

- Problem
  - Exceptions may occur in the catch statement
  - Leaving no option to handle exception
- Solution: using RAII
  - Define a class and a destructor to release file handle
- Let’s rewrite a class to handle file exception

Resource Acquisition Is Initialization (RAII)

```
class FileHandler {
      priviate:
                 FILE* m_fileHandle;
      public:
                FileHandler(const char* fileName, const char* mode) {
                      m_fileHandle = fopen(fileName, mode);
                      if (m_fileHandle == NULL) throw “Unable to open file!”
                }

                ~FileHandler() {
                      fclose(m_fileHandle); // this is a very important part
                }

                writeFile(char* text) {
                     if (fputs(text, m_fileHandle) == EOF ) {
                               throw “Error when writing to file” ;
                      }
                }
};
```

Practice

- Define a division by zero exception
- Write a method to try and catch the division zero exception
- Revise the exception to allow one exception code and a message

- Define exceptions for file input and output errors
  - IOException
    - FileNotFoundException
    - FileOutputException
    - FileInputException