

# Exception Handling

Nguyễn Văn Khiết

# Nội dung

- Khái niệm exception
- throw
- try ... catch
- Lớp std::exception

# Exception

- Exception là các lỗi, ngoại lệ xảy ra trong chương trình vào thời điểm thực thi.
- C++ cho phép điều khiển quá trình xử lý các ngoại lệ này:
  - Phát sinh ngoại lệ (throw)
  - Bắt các ngoại lệ có thể xảy ra và xử lý (catch)
- → bảo đảm chương trình được thực thi đúng, không bị crash.

# Throw exception

- Ta có thể tường minh phát sinh một ngoại lệ khi chương trình gặp các điều kiện không phù hợp.
- Dùng lệnh throw để tường minh phát sinh ngoại lệ.
- VD:

```
double a = 10, b = 0;
if (b==0)
    throw "division by zero";
a = a/b;
```

# Throw exception

- Khi một exception được throw, nếu ta không thực hiện bắt lại (catch), chương trình sẽ dừng ngay tại đó.

# Khối try... catch

```
try {  
    // code  
}  
catch (int param) {  
    //...  
}  
catch (char param)  
{  
    //...  
}  
catch (...)  
{  
    //...  
}
```

# Khối try ... catch

- VD:

```
double a = 10, b = 0;  
try{  
    if (b==0)  
        throw "division by zero";  
    a = a/b;  
}  
catch (const char* s){  
    cout << "Error: "<< s << endl;  
}
```

# Khối try ... catch

- Khối try...catch... cho phép quản lý các ngoại lệ trong đoạn code nằm trong khối try.
- Các ngoại lệ tùy thuộc vào kiểu dữ liệu sẽ được xử lý trong khối catch tương ứng.
- Có thể viết nhiều khối catch ứng với các ngoại lệ khác nhau
- Khối catch (...) sẽ xử lý tất cả các ngoại lệ còn lại chưa được catch trước đó

# Khối try ... catch

- Khi một exception được throw trong một hàm A, nếu hàm A không catch exception này lại, thì hàm B (là hàm gọi A) có thể catch exception này, và cứ tiếp tục cho đến hàm main

# Khối try ... catch

```
void a()
{
    double a = 10, b = 0;
    try{
        if (b==0)
            throw "division by zero";
        a = a/b;
    }
    catch (int i){
        cout << "Error: " << i << endl;
    }
}
void b()
{
    try{
        a();
    }
    catch (const char* s){
        cout << "Error: " << s << endl;
    }
}

int main(){
    b();
}
```

# Khối try ... catch

- Từ hàm A, khi catch một exception, ta có thể throw lại exception này cho hàm B là hàm gọi nó.
- Cú pháp: **throw;**

# Khối try ... catch

```
void a()
{
    double a = 10, b = 0;
    try{
        if (b==0)
            throw "division by zero";
        a = a/b;
    }
    catch (const char* s){
        throw;
    }
}
void b()
{
    try{
        a();
    }
    catch (const char* s){
        cout << "Error: "<< s << endl;
    }
}
```

# Lớp std::exception

- Là lớp cơ sở để ta định nghĩa các lớp mới khi tự tạo ra các exception.
- Lớp này có một hàm ảo (virtual) là what() để mô tả exception
- VD:

```
class WaterTooHotException : public exception {  
public:  
    virtual const char* what() const noexcept {  
        return "The temperature is greater than 100";  
    }  
};
```

# Lớp std::exception

```
void boilwater(int temperature)
{
    if (temperature > 100)
    {
        WaterTooHotException wthe;
        throw wthe;
    }
    //...
}

int main() {
    try{
        boilwater(102);
    }
    catch (WaterTooHotException& e)
    {
        cout<<"Error: " <<e.what()<<endl;
    }
    //...
}
```