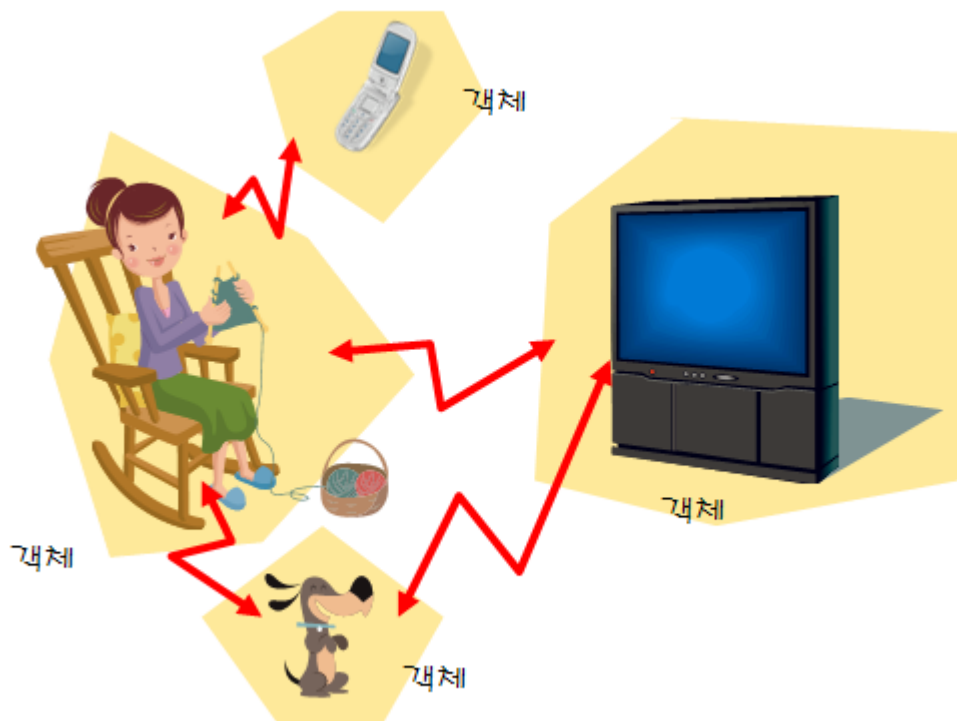




*C++ Espresso*

## 제11장 예외 처리





# 이번 장에서 학습할 내용



- 예외 처리의 개념
- 예외 처리기 구조
- 예외 전달

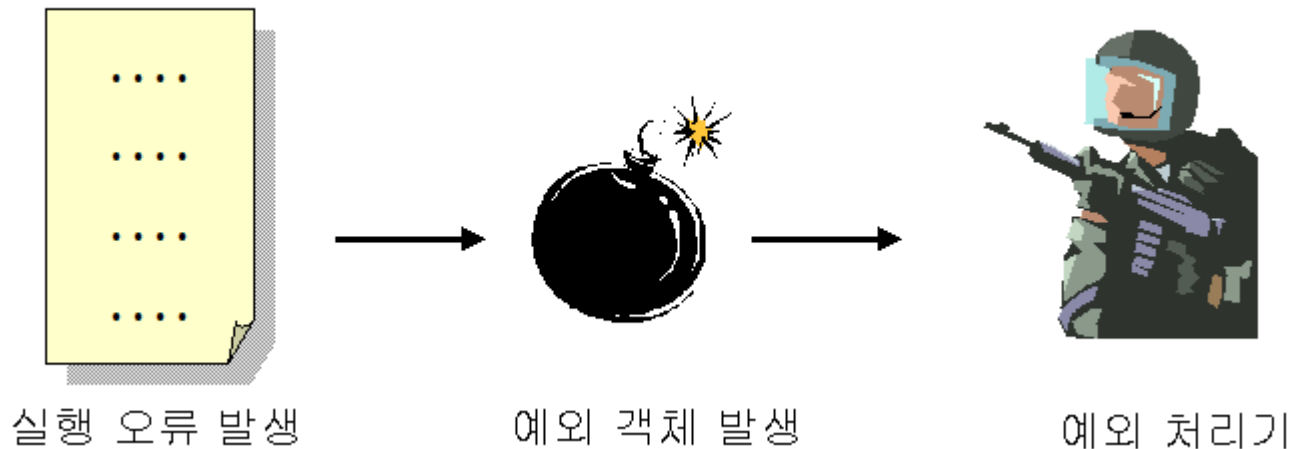
예외는 오류가  
발생하더라도  
오류를  
우아하게  
처리하게  
합니다.





# 예외란?

- 예외(exception): 잘못된 코드, 부정확한 데이터, 예외적인 상황에 의하여 발생하는 오류
- > 프로그램에서 발생할 수 있는 예외 상황은 무엇이 있을까?
- 0으로 나누는 것과 같은 잘못된 연산
  - 배열의 인덱스가 크기를 벗어나는 경우
  - 디스크에서 하드웨어 에러





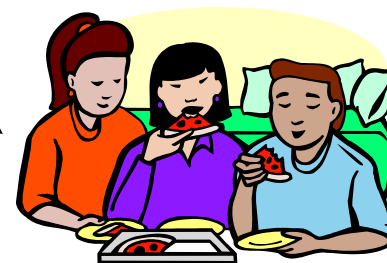
# 예외는 처리하는 것이 좋다.



그림 11.1 예외 처리의 개요



# 피자 나누기 프로그램





# 피자 나누기 프로그램



```
#include <iostream>
using namespace std;
int main()
{
    int pizza_slices = 0;
    int persons = -1;
    int slices_per_person=0;

    cout << "피자 조각수를 입력하시오: ";
    cin >> pizza_slices;
    cout << "사람수를 입력하시오: ";
    cin >> persons;
    slices_per_person = pizza_slices / persons;
    cout << "한사람당 피자는" << slices_per_person << "입니다." << endl;

    return 0;
}
```

Q. 여기서 발생할 수 있는 예외는?

A. - 피자 조각 수가 음수  
- 사람수가 음수  
- 사람수가 0인 경우



피자 조각수를 입력하시오: 12  
사람수를 입력하시오: 4  
한사람당 피자는 3입니다.  
계속하려면 아무 키나 누르십시오 ...



# 전통적인 오류 처리 방식

- **If-else**을 사용하여 조건을 검사
- 정상적인 코드와 오류 처리 분리가 어려움

```
if (pizza_slices < 0) {  
    cout << "피자조각이 음수임";  
} else if (pizza_slices == 0) {  
    cout << "피자조각이 0임";  
} else {  
    if (persons == 0) {  
        cout << "사람이 0명입니다. "<< endl;  
    } else if (persons < 0) {  
        cout << "사람이 음수입니다. " << endl;  
    } else {  
        slices_per_person = pizza_slices / persons;  
        cout << "한사람당 피자는" << slices_per_person <<  
            "입니다." << endl;  
    }  
}
```



# 예외 처리기

## \* try- throw- catch

```
try{           // try 블록은 예외가 발생할 수 있는 블록

    .....
    예외를 발견한다면 {
        throw XXX;
    } // throw는 예외를 던진다. 예외 값을 던진다.
      예) throw 3; // 예외 값 3을 던짐
      예) throw "empty stack"; // char* 타입의 문자열 예외를 던짐
}
catch(처리할 예외 파라미터 선언){
    예외 처리문
}
// catch 블록은 try블록에서 예외가
// 발생하면 처리
// 예외 파라미터는 한 개만 선언 가능하다.
```





# 예외 처리기

다음 프로그램의 실행 결과를 예상해보자~!

```
try {  
    throw 1;  
}  
catch (int x) {  
    cout << x;  
}
```

C:\Windows\system32\cmd.exe

1계속하려면 아무 키나 누르십시오 . . .

```
try {  
    throw 3;  
}  
catch (int x) {  
    try {  
        cout << x;  
        throw "aa";  
    }  
    catch (char* p) {  
        cout << p;  
    }  
}
```

C:\Windows\system32\cmd.exe

3aa계속하려면 아무 키나 누르십시오 . . .



# 피자 나누기 프로그램

```
int main()
{
    int pizzaSlices = 0;
    int persons = 0;
    int slicesPerPerson = 0;
    try {
        cout << "피자 조각 수를 입력하시오:";
        cin >> pizzaSlices;
        cout << "사람수를 입력하시오: ";
        cin >> persons;

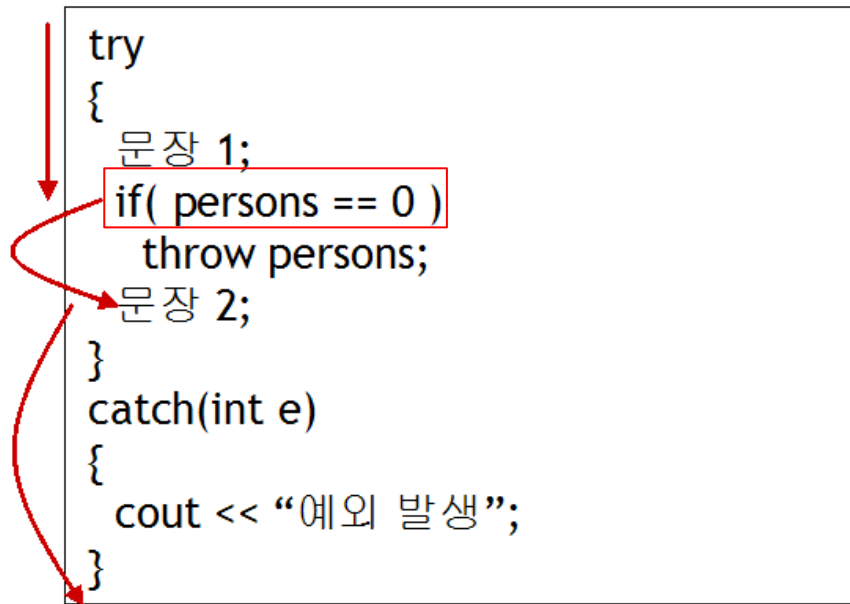
        if (persons == 0)
            throw persons;
        slicesPerPerson = pizzaSlices / persons;
        cout << "한사람당 피자는 " << slicesPerPerson << "입니다." << endl;
    }
    catch (int e){
        cout << "사람이" << e << "명 입니다." << endl;
    }
    return 0;
}
```

피자 조각 수를 입력하시오:12  
사람수를 입력하시오: 4  
한사람당 피자는 3입니다.  
계속하려면 아무 키나 누르십시오 . . .

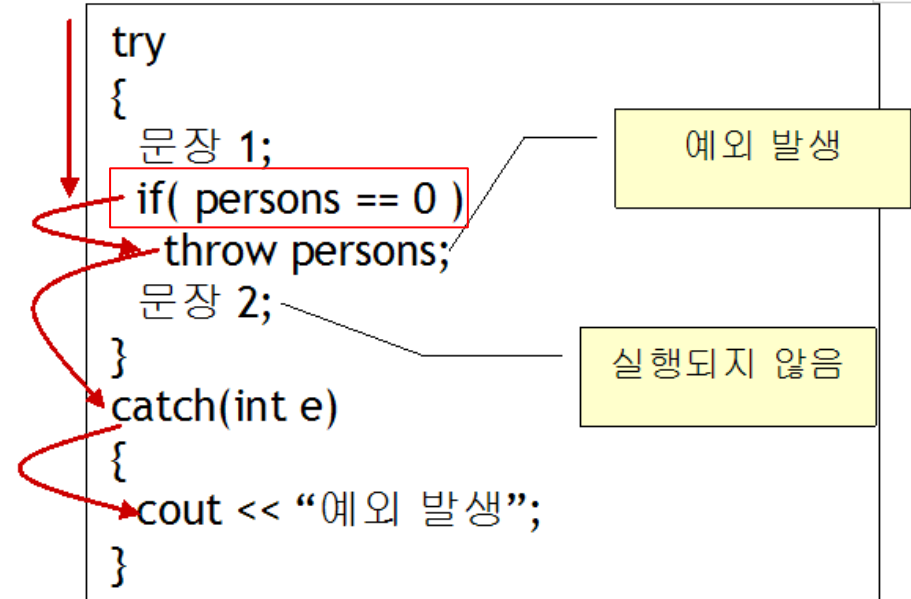
피자 조각 수를 입력하시오:12  
사람수를 입력하시오: 0  
사람이0명 입니다.  
계속하려면 아무 키나 누르십시오 . . .



# try/catch 블록에서의 실행 흐름



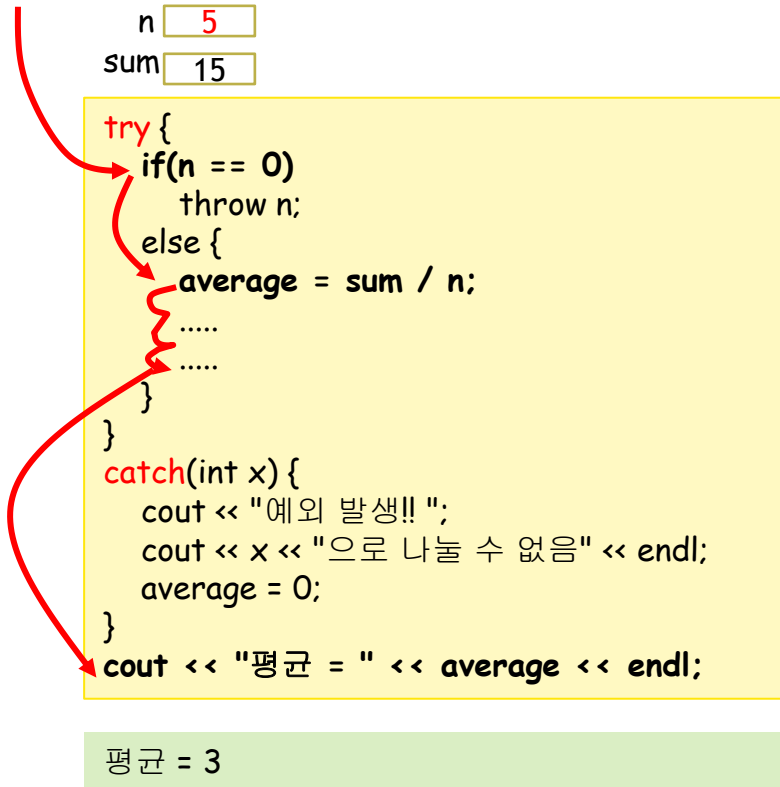
예외가 발생하지 않은 경우



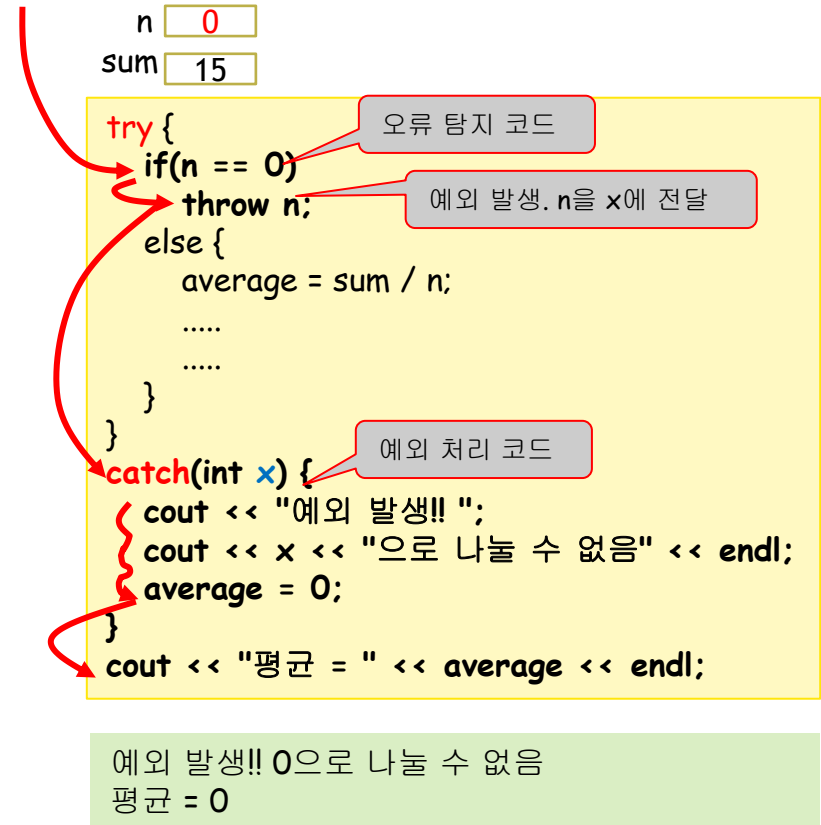
예외가 발생한 경우



# try/catch 블록에서의 실행 흐름



(a) 예외가 발생하지 않은 경우



(b) 예외가 발생한 경우



# catch 블록의 매개 변수

```
try
{
    문장 1;
    if( persons == 0 )
        throw persons;
    문장 2;
}
catch(int e)
{
    cout << “예외 발생”;
}
```

예외 처리기의 매개 변수

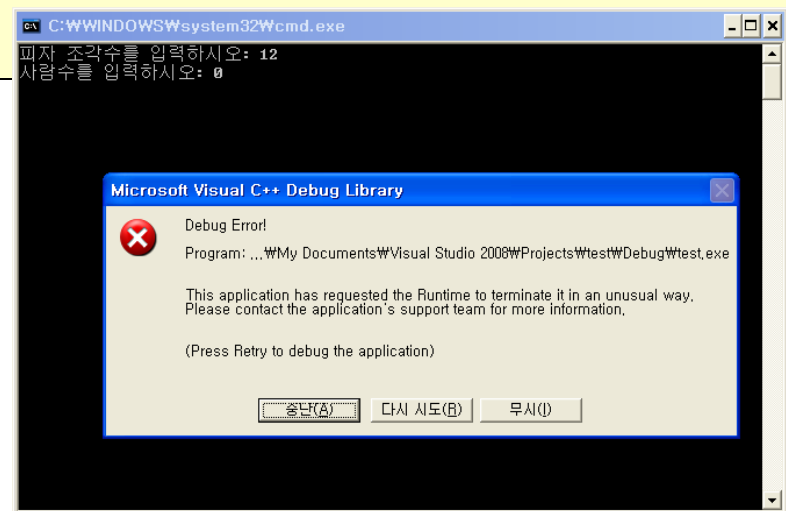


# 타입이 일치되는 예외만 처리



```
try
{
    int person =0;
    ...
    if (persons == 0)
        throw persons;
    ...
}
catch(char e)
{
    cout << "사람이 " << e << " 명 입니다. "<< endl;
}
```

타입이 일치하지 않음





# 함수를 통한 예외 전달

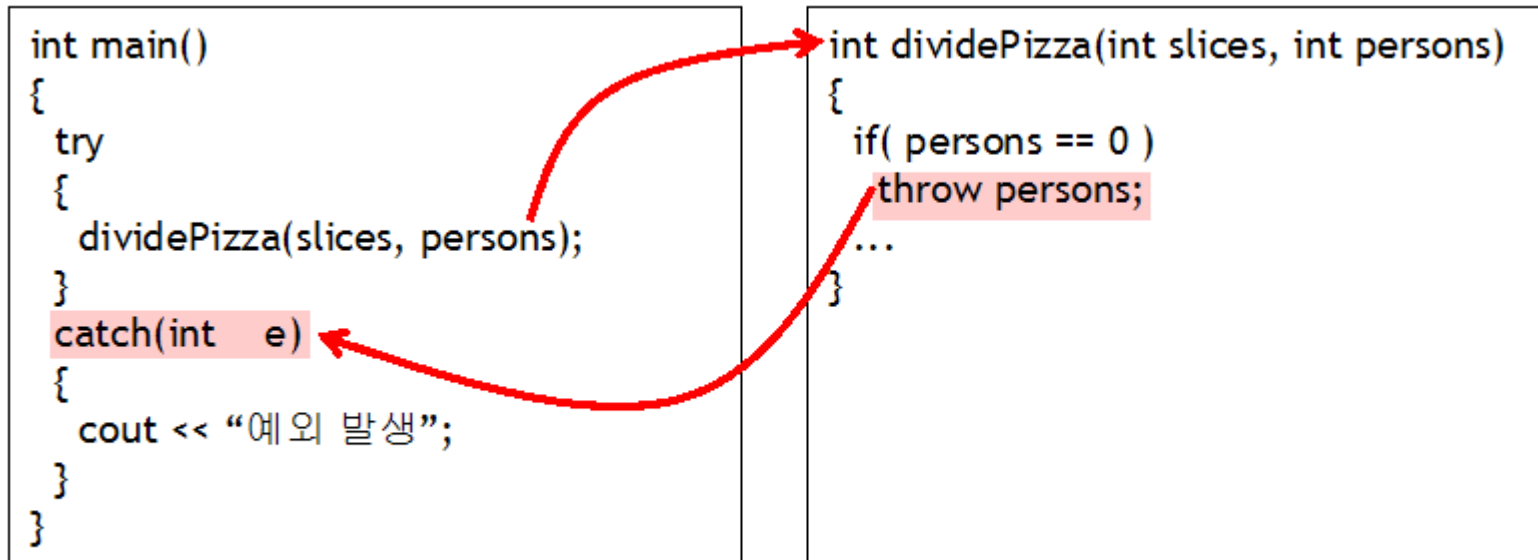


그림 15.5 예외는 함수를 넘어서 전달될 수 있다.



# 예외 전달

# 앞의 예제에 다음과 같이 피자의 조각 수를 계산하여 반환하는 함수를 추가해보자 - **인수로 받은 person이 0인 경우 예외를 발생시키도록 한다.**

```
int main()
{
    int pizzaSlices = 0;
    int persons = 0;
    int slicesPerPerson = 0;
    try {
        cout << "피자 조각 수를 입력하시오:";
        cin >> pizzaSlices;
        cout << "사람수를 입력하시오: ";
        cin >> persons;

        slicesPerPerson = DividePizza(pizzaSlices, persons);
        cout << "한사람당 피자는 " << slicesPerPerson << "입니다." << endl;
    }
    catch (int e){
        cout << "사람이" << e << "명 입니다." << endl;
    }
    return 0;
}
```

```
int DividePizza(int pizzaSlice, int person)
{
    if (person == 0)
        throw person;
    return pizzaSlice / person;
}
```





# 퀴즈

예외가 발생할 가능성이 있는 코드를 ( **try** ) 블록으로 감싸고, 예외를 탐지하는 조건문에서 예외가 탐지되면 ( **throw** ) 문을 이용하여 예외 발생을 알린다. 발생한 예외는 ( **catch** )블록에서 처리된다.

```
int getCubVolume(int a) {  
    try{  
        if(a <=0) throw a;  
        else return a*a*a;  
    }  
    catch(int x){  
        cout << "육면체의 변은 양수이어야 한다." << endl;  
        return 0;  
    }  
}
```

(1) `int n = getCubVolume(5);`에서 예외가 발생하는가? `n`의 값은?

(2) `int n = getCubVolume(-5);`에서 예외가 발생하는가? `n`의 값은?



# Q & A

