



---

# C++

---

#12: (Exception Handling)

2007. 5. 30.

:

E-mail: [jaesoo27@kut.ac.kr](mailto:jaesoo27@kut.ac.kr)

1

- 
- - C++
  - stack unwinding( )
  - [ ]
  - [ ]
  - [ ]new

# 1.

- - 
  - 가 !

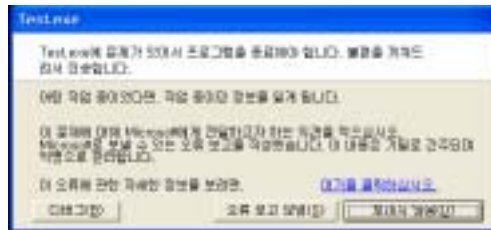
```
int main(void)
{
    int a, b;
    cout<<"          : ";
    cin>>a>>b;

    cout<<"a/b      : "<<a/b<<endl;
    cout<<"a/b      : "<<a%b<<endl;
    return 0;
}
```



# 1.

- : 4 0 , ( ) 0



:



# 1.

- 
- 

```
int main(void)
{
    int a, b;

    cout<<"          : ";
    cin>>a>>b;
    if(b==0){
        cout<<"      !"<<endl;
    }
    else {
        cout<<"a/b      : "<<a/b<<endl;
        cout<<"a/b      : "<<a%b <<endl;
    }
    return 0;
}
```



## 2. C++

### ■ try, catch, throw

### ■ try

```
try {
    /*          */
}
```

### ■ catch

```
catch(          ) {
    /*          가          */
}
```



## 2. C++

### ■ try & catch

```
try {  
    /*          */  
}  
catch(          ){  
    /*          가          */  
}
```

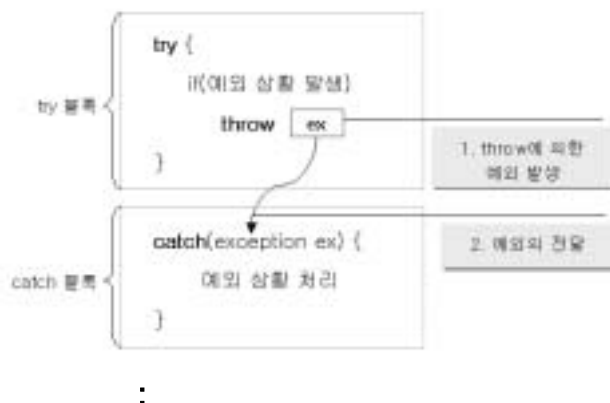
### ■ throw

```
throw ex; // ex 가 “ ” .
```



## 2. C++

### ■ throw , try catch .



## 2. C++

```
int main(void)
{
    int a, b;

    cout<<"          : ";
    cin>>a>>b;

    try{
        if(b==0)
            throw b;
        cout<<"a/b      : "<<a/b<<endl;
        cout<<"a/b      : "<<a%b<<endl;
    }
    catch(int exception){
        cout<<exception<<"      ."<<endl;
        cout<<"          !      ."<<endl;
    }

    return 0;
}
```



## 2. C++

```
int main(void)
{
    int a, b;

    cout<<"          : ";
    cin>>a>>b;

    try{
        cout<<"try block start"<<endl;

        if(b==0)
            throw b;
        cout<<"a/b      : "<<a/b<<endl;
        cout<<"a/b      : "<<a%b<<endl;

        cout<<"try block end"<<endl;
    }
    catch(int exception){
        cout<<"catch block start"<<endl;

        cout<<exception<<"      ."<<endl;
        cout<<"          !      ."<<endl;
    }

    cout<<"THANK YOU!"<<endl;
    return 0;
}
```

- 가 try
- catch

### 3.

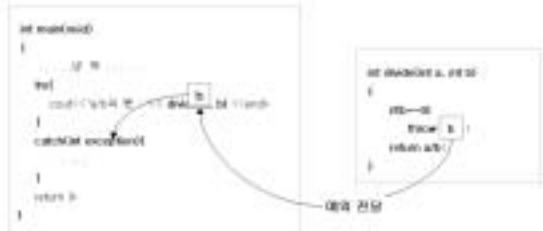
```
int divide(int a, int b) // a/b

int main(void)
{
    int a, b;

    cout<<"          : ";
    cin>>a>>b;

    try{
        cout<<"a/b      : "<<divide(a, b)<<endl;
    }
    catch(int exception){
        cout<<exception<<"      ."<<endl;
        cout<<" |          ."<<endl;
    }
    return 0;
}

int divide(int a, int b)
{
    if(b==0)
        throw b;
    return a/b;
}
```



### 3.

```
#include <iostream>
using std::endl;
using std::cout;
using std::cin;

void fct1();
void fct2();
void fct3();

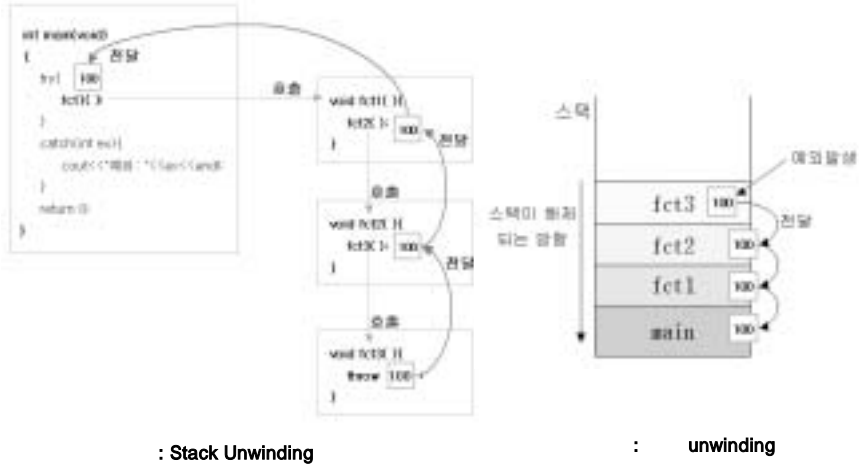
int main(void)
{
    try{
        fct1();
    }
    catch(int ex){
        cout<<"          : "<<ex<<endl;
    }
    return 0;
}

void fct1(){
    fct2();
}
void fct2(){
    fct3();
}
void fct3(){
    throw 100;
}
```

- 가
- 가
- 가
- -> Stack Unwinding



### 3. ( )



### 3. ( )

- )가
  - `stdlib.h` abort
  - `unhandel1.cpp`, `unhandel2.cpp`
- - 
  - 가 abort

```

int fct(double d) throw (int, double, char *)
{
    .....
}
int fct(double d) throw () //
{
    ....
}
    
```



### 3.

```
// unhandle1.cpp
#include <iostream>
using std::endl;
using std::cout;
using std::cin;

int divide(int a, int b); // a/b

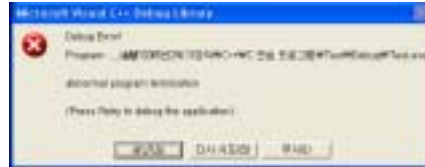
int main(void)
{
    int a, b;

    cout<<"          : ";
    cin>>a>>b;
    cout<<"a/b      : "<<divide(a, b)<<endl;

    return 0;
}

int divide(int a, int b)
{
    if(b==0)
        throw b;
    return a/b;
}
```

- 가 ,
- 가
- `stdlib.h` `abort`
- 가



: `abort`

### 3.

```
// unhandle2.cpp
#include <iostream>
using std::endl;
using std::cout;
using std::cin;

int divide(int a, int b); // a/b

int main(void)
{
    int a, b;

    cout<<"          : ";
    cin>>a>>b;
    try{
        cout<<"a/b      : "<<divide(a, b)<<endl;
    }
    catch(char exception){
        cout<<exception<<"          .<<endl;
        cout<<"          !          .<<endl;
    }

    return 0;
}

int divide(int a, int b)
{
    if(b==0)
        throw b;
    return a/b;
}
```

- Try ~ Catch `char`
- > `abort`



### 3.

- try, catch
- catch\_understand.cpp

```
int main(void)
{
    int num;

    cout<<"input number: ";
    cin>>num;

    try{
        if(num>0)
            throw 10; // int
        else
            throw 'm'; // char
    }
    catch(int exp){
        cout<<"int          "<<endl;
    }
    catch(char exp){
        cout<<"char          "<<endl;
    }
    return 0;
}
```

- Try catch
- Try ~ catch
- catch



<

>

```
#include <iostream>
using std::endl;
using std::cout;
using std::cin;

char* account="1234-5678"; //
int sid=1122; //
int balance=1000; //

class AccountExpt
{
    char acc[10];
    int sid;
public:
    AccountExpt(char* str, int id){
        strcpy(acc, str);
        sid=id;
    }
    void What(){
        cout<<"          "<<endl;
        cout<<"          "<<endl;
    }
};
```

```
int main(void)
{
    char acc[10];
    int id;
    int money;

    cout<<"          : ";
    cin>>acc;
    cout<<"          4          : ";
    cin>>id;

    if(strcmp(account, acc) || sid!=id)
        throw AccountExpt(acc, id);

    cout<<"          : ";
    cin>>money;
    if(balance<money) throw money;

    balance-=money;
    cout<<"          : "<<balance<<endl;
}

return 0;
}
```



```
//
#include <iostream>
using std::endl;
using std::cout;
using std::cin;

char* account="1234-5678"; //
int sid=1122; //
int balance=1000; //

class AccountExpt
{
    char acc[10];
    int sid;
public:
    AccountExpt(char* str, int id){
        strcpy(acc, str);
        sid=id;
    }
    void What(){
        cout<<"          : "<<acc<<endl;
        cout<<"          : "<<sid<<endl;
    }
};

int main(void)
{
    char acc[10];
    int id;
    int money;
```

```
try{
    cout<<"          : ";
    cin>>acc;
    cout<<"          4          : ";
    cin>>id;
    if(strcmp(account, acc) || sid!=id)
        throw AccountExpt(acc, id);
}

catch(AccountExpt& expt){
    cout<<"          " <<endl;
    expt.What();
}

try{
    cout<<"          : ";
    cin>>money;
    if(balance<money)
        throw money;

    balance -=money;
    cout<<"          : "<<balance<<endl;
}

catch(int money){
    cout<<"          : "<<money - balance<<endl;
}

return 0;
}
```

- ?
- Try 가 -> Try



```
< >

#include <iostream>
using std::endl;
using std::cout;
using std::cin;

char* account="1234-5678"; //
int sid=1122; //
int balance=1000; //

class AccountExpt
{
    char acc[10];
    int sid;
public:
    AccountExpt(char* str, int id){
        strcpy(acc, str);
        sid=id;
    }
    void What(){
        cout<<"          : "<<acc<<endl;
        cout<<"          : "<<sid<<endl;
    }
};

};
```

```
int main(void)
{
    char acc[10];
    int id;
    int money;

    try{
        cout<<"          : ";
        cin>>acc;
        cout<<"          4          : ";
        cin>>id;

        if(strcmp(account, acc) || sid!=id) throw AccountExpt(acc, id);
        cout<<"          : ";
        cin>>money;
        if(balance<money) throw money;

        balance -=money;
        cout<<"          : "<<balance<<endl;
    }

    catch(int money){
        cout << "          " << money - balance << endl;
    }

    catch(AccountExpt& expt){
        cout << "          " << endl;
        expt.What();
    }

    return 0;
}
```



```

//
#include <iostream>
using std::endl;
using std::cout;
using std::cin;

char* account="1234-5678"; //
int sid=1122; //
int balance=1000; //

class AccountExpt
{
    char acc[10];
    int sid;
public:
    AccountExpt(char* str, int id){
        strcpy(acc, str);
        sid=id;
    }
    void What(){
        cout<<"          : "<<acc<<endl;
        cout<<"          : "<<sid<<endl;
    }
};

int main(void)
{
    char acc[10];
    int id;
    int money;

```

```

try{
    cout<<"          : ";
    cin>>acc;
    cout<<"          4          : ";
    cin>>id;
    if(strcmp(account, acc) || sid!=id)
        throw AccountExpt(acc, id);

    cout<<"          : ";
    cin>>money;
    if(balance<money)
        throw money;

    balance -=money;
    cout<<"          : "<<balance<<endl;
}

catch(int money){
    cout<<"          : "<<money - balance<<endl;
}

catch(AccountExpt& expt){
    cout<<"          "<<endl;
    expt.What();
}

return 0;
}

```



< >

- 
- 



&

---

*Thank You !*

