# 프로그래밍 언어 (2)
# 실습

참고 문헌 : IT강좌 프론티어 (www.freelec.com)

# 다중 상속

- **ex1)**

```
#include<iostream>
#include<string>

using namespace std;

class Employee{
private:
 int id;
 string name;

public:

 Employee(int , string);
 ~Employee();
 virtual int CalculatePay() const;
 virtual void Print() const;

};

Employee::Employee(int i, string n) : id(i),name(n){
}
Employee::~Employee(){
}
void Employee::Print() const{
 cout << "Id:" << id << " Name : " << name << " 급여 : " << CalculatePay() << endl;
}
int Employee::CalculatePay() const
{
 return 0;
}
```

# 다중 상속

- **ex1)** 계속

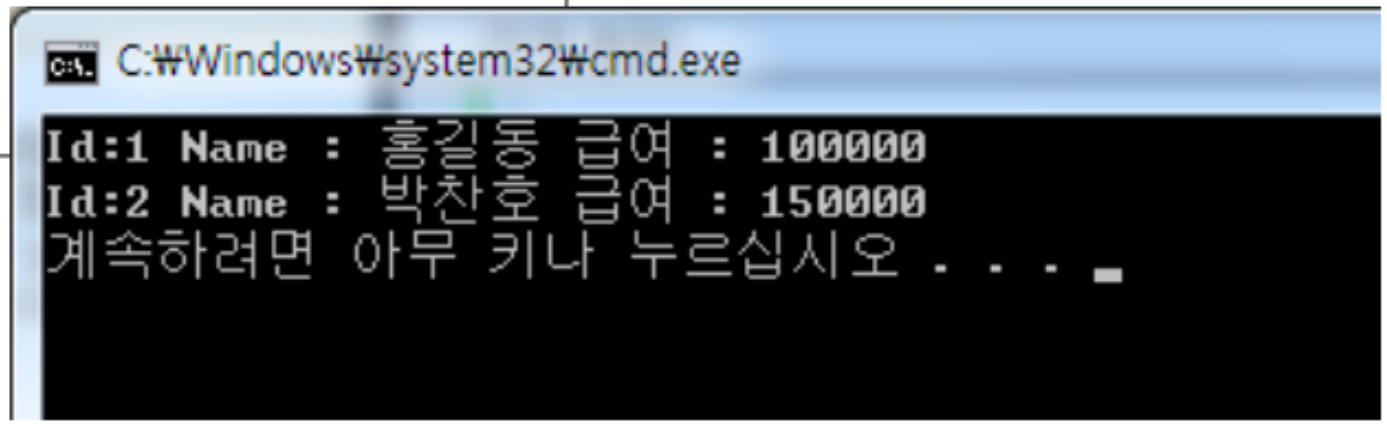
```
class SalariedEmp :public Employee{
private:
 int salary;
public:

 SalariedEmp(int i , string n, int s) : salary(s),Employee(i,n){

 }

 int CalculatePay() const{
  return salary;
 }

};
```

```
class HourlyEmp : public Employee{
private:
 int hours;
 int Payperhours;
public:

 HourlyEmp(int i , string n, int h, int p) : hours(h),Payperhours(p), Employee(i,n){

 }

 int CalculatePay() const{
  return hours * Payperhours ;
 }

};
```

# 다중 상속

- **ex1)** 계속

```
int main(){
 Employee* p[2] ;

 p[0] = new SalariedEmp(1, "홍길동", 100000);
 p[1] = new HourlyEmp(2,"박찬호",30,5000);

 for (int i = 0 ; i < 2 ; i ++){
  p[i]->Print();
 }

 for (int i = 0 ; i < 2 ; i ++){
   delete p[i];
 }

 return 0;
}
```



# 다중 상속

- **ex2)**

```
#include<iostream>
#include<string>

using namespace std;

class Person
{
private:
 int id;
 string name;
public:
 Person(){
 }

 Person(int i, string n) : id(i),name(n) {
  cout << "Person Class // " << " id = "<< i <<" name = " << n << endl;
 }

};
```

# 다중 상속

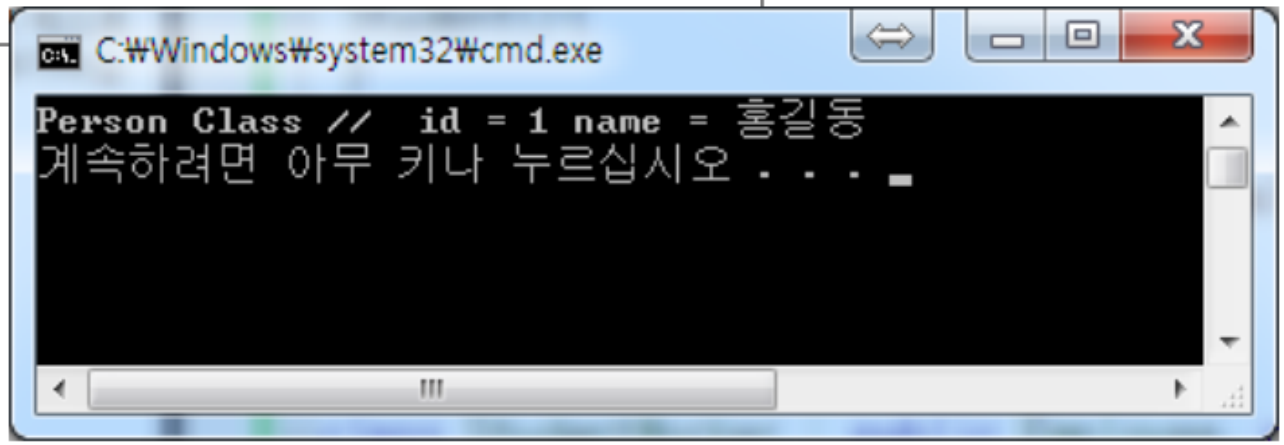
- **ex2)** 계속

```
class Employee : virtual public Person{
private:
 int salary;
public:
 Employee(){
 }

 Employee(int id, string name, int sa): salary(sa),Person(id,name) {
 }
};

class Student :virtual public Person{
private:
 string major;

public:
 Student(){
 }

 Student(int id, string name, string ma): major(ma),Person(id,name) {
 }
};
```

# 다중 상속

- **ex2)** 계속

```
class StudentWorker : public Employee, public Student{

public :
 StudentWorker(int id, string name, int salary, string major)
  :Student(id,name,major),Employee(id,name,salary),
  Person(id,name) {
 }

};

int main(){

 StudentWorker sw(1,"홍길동",1000000,"수학과");

}
```

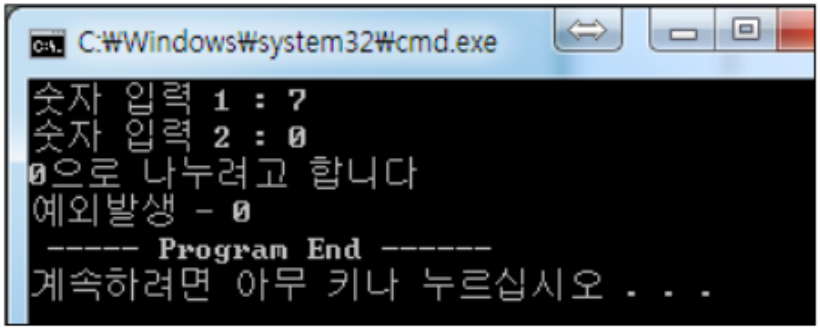


# 예외 처리

- **ex3)**

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

int main()
{
 int a, b;
 int div;

 try
 {
  cout << "숫자 입력 1 : " ;
  cin >> a;

  cout << "숫자 입력 2 : " ;
  cin >> b;

  if (b == 0)
  {
   throw 0;
  }

  div = a/b;

  cout << "나눈 값은 " << div << endl;
 }
```

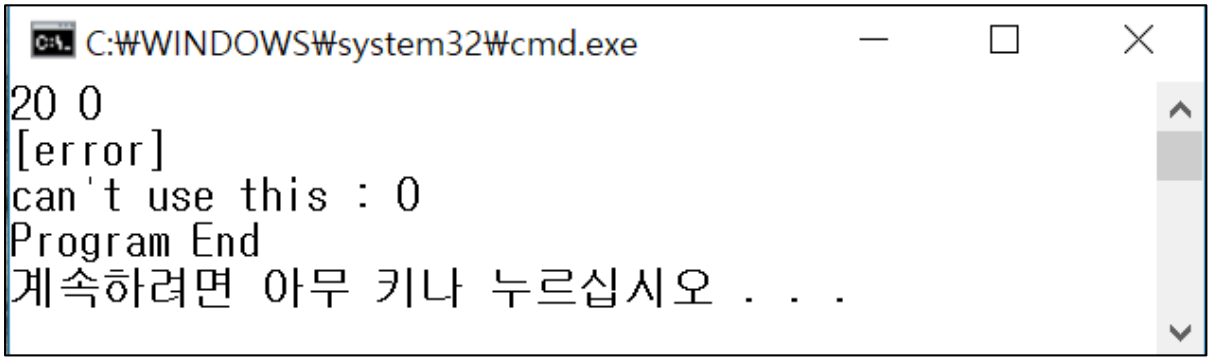
```
 catch(int i)
 {
  if (i == 0)
  cout << "0으로 나누려고 합니다" << endl;

  cout << "예외발생 - " << i << endl;
 }
 catch(char *s)
 {
  cout << "예외발생 - " << s << endl;
 }
 catch(...) //모든 예외  - 맨뒤에 ..
 {
  cout << "예외발생" << endl;
 }
 cout << " ----- Program End ------" << endl;
}
```



# 예외 처리

- **ex4)**

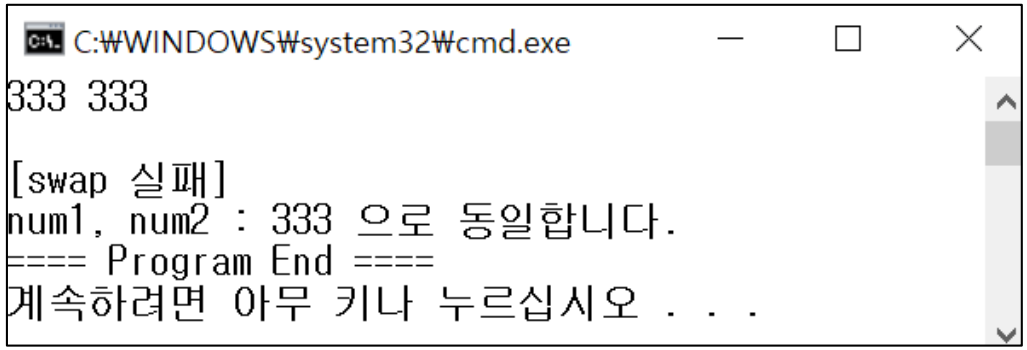
```
#include <iostream>
using namespace std;
int main(void){
    int a, b;
    try{//try에서 예외가 있는지 탐색.
        cin >> a;
        cin >> b;
        if(b<=0){//예외 발생시,
            throw b; //throw를 통해 변수 b를 catch의 파라미터로 보냄.
        }
        cout << "[work]" << endl;
        cout << "a : " << a << " , b : " << b << endl;
        cout << "a/b : " << a/b <<endl;
    }catch(int expn){//throw에서 보낸 b를 인자를 expn으로 받는다.
        cout << "[error]" << endl;
        cout << "can't use this : " << expn << endl;
    }
    cout << "Program End" << endl;
    return 0;
}
```



# 예외 처리

- **ex5)**

```
#include <iostream>
using namespace std;
void swap(int &a, int &b){
    int tmp;
    if(a == b) throw a;  //호출된 함수 내부에서 예외 처리를 하여 throw를 통해 catch에 보낸다.

    tmp = a;
    a = b;
    b = tmp;
}
int main(void){
    int num1;
    int num2;
    try{
        cin >> num1;
        cin >> num2;
        //try 내부에서 swap 함수 호출.
        swap(num1, num2);
        cout << "₩n[swap 완료]" << endl;
        cout << "num1 : " << num1 << ", num2 : " << num2 << endl;

    }catch (int expn){ // throw를 통해 변수를 인자값으로 받는다.
        cout << "₩n[swap 실패]" << endl;
        cout << "num1, num2 : " << expn << " 으로 동일합니다." << endl;
    }
    cout << "==== Program End ====" <<endl;
    return 0;
}
```



# 참고문헌


- https://modoocode.com/210
- https://wikidocs.net/22468
- http://algamza.blogspot.com/2016/03/c-operator-overloading.html
- http://tcpschool.com/cpp/cpp_operatorOverloading_intro
- https://blockdmask.tistory.com/55

# Q & A

참고 문헌 : IT강좌 프론티어 (www.freelec.com)