

프로그래밍입문(2) 실습

14주차

예외처리

- 1)다음은 입력한 두 정수의 곱셈과 나눗셈 연산을 출력하는 프로그램이다. 출력결과를 참고하여 오류가 나지 않게 수정하시오.
- multi(), division() 함수

```
void multi(int a=0, int b=0){
    cout<< "multi result : " << a*b << endl;
}

void division(int a=0,int b=1){
    cout << "division result : " << a/b << endl;
}
```

- 출력결과

출력결과
계좌번호:110 계좌잔액 : 1200

예외처리

- 1) Solution

```
#include <iostream>
using namespace std;
void multi(int a=0, int b=0){
    cout << "multi result : " << a*b << endl;
}
void division(int a=0,int b=1){
    try{
        if( b==0 ){
            throw 0;
        }
        cout << "division result : " << a/b << endl;
    }
    catch(int exception){
        cout << exception << "으로 나눌 수 없습니다 " << endl;
    }
}
int main(){
    int val_1, val_2;
    cout << "두 개의 숫자 입력 : ";
    cin >> val_1 >> val_2;
    multi(val_1, val_2);
    division(val_1,val_2);
    return 0;
}
```

예외처리

- 2) **main** 함수와 출력결과를 보고 다음 조건에 맞게 프로그램을 작성하시오
- **swich, case**를 이용
- **val**값이 1이면 “**Bye**”, 2이면 **10.5**, 3이면 ‘**a**’가 출력

출력결과

1
Bye

- 출력결과

출력결과

2
10.5

출력결과

3
a

예외처리

- main

```
int main(){  
  
    int val;  
    cin >> val;  
    try{  
        func1(val);  
    }  
    catch(const char* s){  
        cout << s << endl;  
    }  
    catch(double d){  
        cout << d << endl;  
    }  
    catch(char c){  
        cout << c << endl;  
    }  
    catch(int i){  
        cout << i << endl;  
    }  
  
}
```

예외처리

- 2) Solution

```
#include <iostream>
#include <string
using namespace std;
void func1(int val){
    switch(val){
        case 1: throw "Bye";
        case 2: throw 10.5;
        case 3: throw 'a';
        default: throw 0;
    }
}
int main(){
    int val;
    cin >> val;
    try{
        func1(val);
    }
    catch(const char* s){
        cout << s << endl;
    }
    catch(double d){
        cout << d << endl;
    }
    catch(char c){
        cout << c << endl;
    }
    catch(int i){
        cout << i << endl;
    }
}
```

템플릿

- 3) 템플릿을 이용하여 다음 조건을 만족시키는 **swap** 함수를 작성하시오
- 함수의 이름은 **swap_value**
- 입력 매개변수가 **int**가 아닌 **double**, **float**이어도 동작해야함

- **main** 함수

```
int main(){  
    int a, b;  
    cin >> a >> b;  
    swap_value(a,b);  
    cout << a << b;  
}
```

- **출력결과**

```
10 20  
2010
```

출력결과

템플릿

- 3) Solution

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

template <typename T>
void swap_value(T &t1, T &t2){

    T temp;

    temp=t1;

    t1=t2;

    t2=temp;

}
int main(){

    int a, b;

    cin >> a >> b;

    swap_value(a,b);

    cout << a << b;

}
```


템플릿

- 4) 다음 함수들은 입력과 출력 자료형마다 나누어 놓은 함수들이다. 같은 동작을 하는 템플릿 함수를 선언하시오

- 여러 **max** 함수

```
short max(short a, short b){
    return a<b? b : a;
}

double max(double a, double b){
    return a<b? b : a;
}

int max(int a, int b){
    return a<b? b : a;
}
```

템플릿

- 4) Solution

```
template <typename T>  
T max( T a, T b )  
{  
    return a < b ? b : a;  
}
```

객체 템플릿

- 5) 다음 main 함수와 출력 결과를 참고하여 **student** 클래스에 학번과 성적을 저장하고, 큰 숫자를 출력하는 **get_Max** 함수를 작성하시오.

- **main** 함수

```
int main(){  
  
    Student<int, float> st1(2019,79.5);  
    Student<int, int> st2(2020,80);  
    getMax(st1.getGrade(), st2.getGrade());  
  
    st1.setGrade(80.5);  
    getMax(st2.getGrade(), st1.getGrade());  
  
}
```

- **출력결과**

출력결과

80
80.5

객체 템플릿

- 5) Solution

```
#include <iostream>
using namespace std;
template <typename T1, typename T2>
class Student{
    T1 std_ID;
    T2 grade;
    public:
    Student(T1 std_ID, T2 grade){
        this->std_ID=std_ID;
        this->grade=grade;
    }
    void setGrade(T2 number){
        grade=number;
    }
    T2 getGrade(){ return grade; }
};
template <typename T3,typename T4>
void getMax(T3 a, T4 b){
    if (a>b){
        cout<< a<< endl;
    }
    else cout << b << endl;
}
int main(){

    Student<int, float> st1(2019,79.5);
    Student<int, int> st2(2020,80);
    getMax(st1.getGrade(), st2.getGrade());

    st1.setGrade(80.5);
    getMax(st2.getGrade(), st1.getGrade());

}
```

참고문헌

- 명품 C++ Programming, 황기태 , 생능출판사, 2018

Q & A