

LAB #9



● 실습 목표 – 함수 템플릿과 클래스 템플릿 이해와 활용

1. 템플릿 함수의 작성 방법 이해
2. 혼합된 매개 변수를 이용하는 템플릿 함수의 작성 방법 이해
3. 함수 템플릿 대 매크로
4. 템플릿 함수의 다중 일반형 키 사용
5. 템플릿 함수에 대한 Specialized Function
6. 템플릿 클래스 활용
7. 템플릿 클래스 상속과 템플릿 Friend 함수

13. Template | Practice 1

• 실습 문제 1 <Section 13-1>

목표 : 템플릿 함수의 작성 방법 이해

- int, double, float 자료형에 대하여 아래와 같이 절대값을 구하는 함수를 하나의 템플릿 함수로 구현하라.

함수 오버로딩 이용

```
int abs_fun(int num){
    if(num<0) num = -num;
    return num;
}

double abs_fun(double num){
    if(num<0) num = -num;
    return num;
}

long int abs_fun(long int num){
    if(num<0) num = -num;
    return num;
}

void main(){
    int a = -5;
    double b = -3.14;
    long int c = -153L;
    cout << "a의 절대값 : " << abs_fun(a) << endl;
    cout << "b의 절대값 : " << abs_fun(b) << endl;
    cout << "c의 절대값 : " << abs_fun(c) << endl;
}
```

함수 템플릿 이용

```
template <class DATATYPE>
DATATYPE abs_Tpl(DATATYPE num)
{
    // 절대값 구하는 코드
}

void main(){
    int a = -5;
    double b = -3.14;
    long int c = -153L;
    cout << "a의 절대값 : " << abs_Tpl(a) << endl;
    cout << "b의 절대값 : " << abs_Tpl(b) << endl;
    cout << "c의 절대값 : " << abs_Tpl(c) << endl;
}
```

output

```
C:\Windows\system32\cmd.exe
-5의 절대값은 -> 5
-3의 절대값은 -> 3
-153의 절대값은 -> 153
계속하려면 아무 키나 누르십시오 . . .
```

13. Template | Practice 1

• 실습 문제 2 <Section 13-1>

목표 : 혼합된 매개 변수를 이용하는 템플릿 함수의 작성 방법 이해

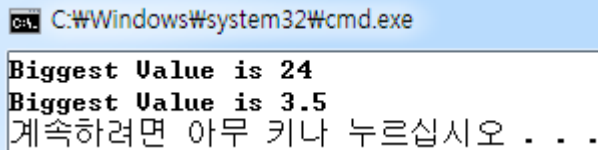
- 하나 이상의 자료형 배열을 받아 최대값을 반환하는 템플릿 함수(biggest)를 작성하라.

```
template <class TYPE>
TYPE biggest (TYPE arr[], int arr_size){
    //(코드 작성)배열과 배열 사이즈를 받아 최대값을 찾는 함수 작성
}

void main()
{
    int ary1[4] = {10, 7, 11, 24};
    cout << "Biggest Value is " << biggest(ary1, 4) << endl;

    float ary2[4] = {3.5f, 1.12f, 3.14f, 0.25f};
    cout << "Biggest Value is " << biggest(ary2, 4) << endl;
}
```

output



```
C:\Windows\system32\cmd.exe
Biggest Value is 24
Biggest Value is 3.5
계속하려면 아무 키나 누르십시오 . . .
```

13. Template | Practice 1

• 실습 문제 3 <Section 13-1>

목표 : 함수 템플릿 대 매크로

- 아래와 같은 매크로 함수를 템플릿 함수로 변경하라.

```
#define MAX(x, y) (((x) > (y)) ? (x) : (y)) //매크로 함수
```

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

//(코드작성) 매크로 함수MAX와 동일한 동작을 수행하는 템플릿 함수MAX_TPL을 작성하시오.

```
void main()
{
    cout << "Begin macro testsWn";
    cout << "Test1 : " << MAX(1,4) << endl;
    cout << "Test2 : " << MAX('A', 'B') << endl;
    cout << "Test3 : " << MAX(4, 'C') << endl;
    cout << "End of macro testsWn";

    cout << "Begin Template testsWn";
    cout << "Test4 : " << MAX_TPL(1,4) << endl;
    cout << "Test5 : " << MAX_TPL('A', 'B') << endl;
    cout << "Test6 : " << MAX_TPL(4, 'C') << endl; /*에러문을 확인하고 그 이유를 생각해보시오*/
    cout << "Test3과 Test6의 예를 통해 매크로 함수가 가지는 문제점을 설명하시오" << endl;
    cout << "End of Template testsWn";
}
```

output

```
C:\Windows\system32\cmd.exe
Begin macro tests
Test1 : 4
Test2 : B
Test3 : 67
End of macro tests
Begin Template tests
Test4 : 4
Test5 : B
Test3과 Test6의 예를 통해 매크로 함수가 가지는 문제점을 설명하시오
End of Template tests
계속하려면 아무 키나 누르십시오 . . .
```

13. Template | Practice 1

• 실습 문제 4 <Section 13-1>

목표 : 템플릿 Search함수를 이용한 다중 일반형 키 사용

- 아래 프로그램은 구조화된 요소를 포함하는 배열에 대한 단순한 순차적 탐색을 위한 템플릿이다. 아래 요구 사항을 만족할 수 있는 테스트 프로그램을 작성하라.
- 구조체 sBox 선언
 - key : 정수형 id
 - age : 정수형
 - salary : 부동 소수형

```
template <class TYPE, class KEY>
int search (TYPE arr[], KEY key, int size)
{
    int index = 0;
    while ( index < size )
    {
        if ( key != arr[index].key )
            index++;
        else
            return index;
    }
    return -1;
}
```

```
void main()
{
    sBox arr[3] = { {5, 32, 3.23f}, {2, 48, 5.18f}, {7, 60, 6.01f} };
    //(코드 작성) 테스트 프로그램
}
```

output

```
C:\Windows\system32\cmd.exe
Key : 2
Age : 48
Salary : 5.18
found at location : 1
계속하려면 아무 키나 누르십시오 . . .
```

13. Template | Practice 1

• 실습 문제 5 <Section 13-1>

목표 : 템플릿 함수에 대한 Specialized Function

아래와 같은 템플릿 max함수는 Fraction과 같은 특정형에 대하여 처리 불가능하다. Fraction에 대한 max함수가 동작할 수 있도록 오버로드된 max함수를 작성하라.

```
template <class TYPE>
TYPE max (TYPE x, TYPE y)
{
    return (x > y) ? x : y ;
}

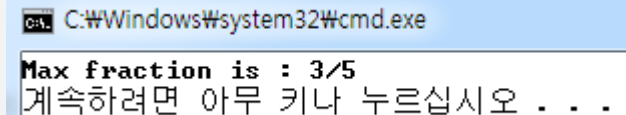
class Fraction{
private:
    int numerator;
    int denominator;
public:
    Fraction(int num, int den) :numerator(num), denominator(den){}
    int compare(Fraction fr2);
    void print() { cout << numerator << "/" << denominator << endl;}
};

int Fraction::compare(Fraction fr2)
{
    if (numerator*fr2.denominator - fr2.numerator*denominator > 0)
        return 1;
    else
        return -1;
}
```

```
void main()
{
    Fraction fr1(3,5);
    Fraction fr2(1,2);

    cout << "Max fraction is : ";
    max(fr1, fr2).print();
}
```

output



```
C:\Windows\system32\cmd.exe
Max fraction is : 3/5
계속하려면 아무 키나 누르십시오 . . .
```

13. Template | Practice 1

• 실습 문제 6 <Section 13-2>

목표 : 템플릿 클래스의 정의와 사용 방법 이해

- 다양한 자료형 배열을 위한 smartAry 클래스를 작성하라.

```
template <class TYPE>
class smartAry
{
private:
    //(코드 작성) 동적으로 할당 받은 배열을 저장할 수 있는 TYPE형 포인터 변수(m_ptr)
    //(코드 작성) 배열 사이즈(arySize)
public:
    smartAry(int size);           //생성자
    ~smartAry();                 //소멸자
    TYPE& operator[] (int idx );  //[ ] 연산자 오버로딩
    int GetSize(); { //(코드작성) }; //배열 길이 반환
};

template <class TYPE>
smartAry<TYPE>::smartAry(int size){    //(코드작성) size 만큼의 TYPE형 동적 배열 할당 및 멤버 변수 초기화 }
//(코드 작성) 소멸자 - 멤버 변수에 할당된 메모리 해제
//(코드 작성) [ ] 연산자 오버로딩 - 매개변수로 입력 받은 인덱스의 TYPE& 반환

void main()
{
    smartAry<int> arr1(4);
    arr1[0] = 1; arr1[1] = 4;
    //(코드 작성) for문을 이용한 arr1의 출력 코드 작성
    //(코드 작성) 동적 할당된 double형 배열을 저장하는 smartAry 객체(arr2) 생성
    arr2[0] = 5.5; arr2[3] = 3.14;
    //(코드 작성) for문을 이용한 arr2의 출력 코드 작성
}
```

output

```
C:\Windows\system32\cmd.exe
1
4
-842150451
-842150451
5.5
-6.27744e+066
-6.27744e+066
3.14
계속하려면 아무 키나 누르십시오 . . .
```

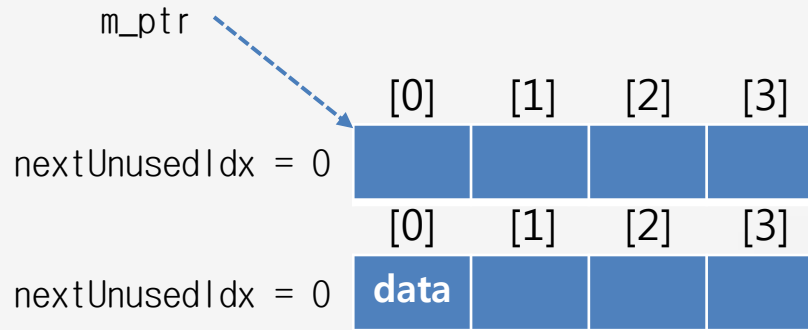
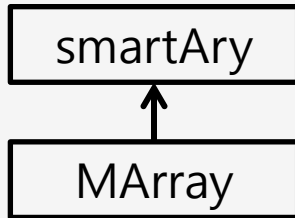

• 실습 문제 7 <Section 13-2>

목표 : 템플릿 클래스 활용(상속)

- 실습 문제4의 smartAry를 상속 받는 MArray 클래스를 아래 요구 사항을 반영하여 작성하라.
 - smartAry의 멤버 변수
 - 데이터 은닉과 상속을 모두 만족할 수 있는 접근 지정자로 변경
 - 멤버 변수
 - 현재 사용 중이지 않은 다음 배열 인덱스 저장 private 변수(nextUnusedIdx) 추가
 - 멤버 함수
 - Marray() : 생성자, smartAry 멤버 변수는 smartAry 생성자를 호출하여 초기화, nextUnusedIdx = 0
 - addElem() : 매개변수로 데이터를 받아 배열에 삽입, nextUnusedIdx 증가, 가득 찬 배열에 데이터 삽입 시 “Error : Array overflow” 메시지 출력
 - isFull() : 배열이 가득 찼는지 여부(true, false) 반환
 - find() : TYPE data를 입력 받아 검색 결과와 해당 원소의 인덱스 값 출력
 - `template <class TYPE> friend void aryPrint(MArray<TYPE>& arr)` : 배열 출력을 위한 함수

13. Template | Practice 1

• 실습 문제 7 <Section 13-2>



addElem(data) 호출 전 nextUnusedIdx = 0

호출 후 nextUnusedIdx = 1

```
void main()
{
    MArray<int> arr1(4);
    for(int i=0; i<arr1.GetSize(); i++)
        arr1.addElem(i*5);
    aryPrint(arr1);
    arr1.find(10);
    cout << "Value at location 3 is : " << arr1.retrieve(3) << endl;

    MArray<double> arr2(4);
    for(int i=0; i<arr2.GetSize(); i++)
        arr2.addElem((i+1)*3.14);
    aryPrint(arr2);
    arr2.find(9.41);
    arr2.addElem(1.15);}
```

output

```
C:\Windows\system32\cmd.exe
[0]0 [1]5 [2]10 [3]15
SUCCESS! Input 10 : index [2]
Value at location 3 is : 15
[0]3.14 [1]6.28 [2]9.42 [3]12.56
FAIL! Input 9.41 : Can not find
Error : Array overflow
계속하려면 아무 키나 누르십시오 . . .
```