# 창의적 소프트웨어 설계 실습 문제 12 – hw12-1

**제출 기한**

12월 4일 화 11:59 PM

1. hw12-1(mkdir hw12-1)라는 폴더를 만들고 GitLab에 push
2. hw12-1 디렉토리에 CMakeLists.txt과 각 문제에서 요구하는 파일들을 작성
3. **‘cmake’ 와 ‘make’명령을 수행하여 숙제가 모두 빌드**
4. 최종 버전을 GitLab에 commit
5. **시간과 파일명, 입력과 출력 방식 반드시 지키기**

**본 실습에서는 function template과 class template의 기본적인 사용법을 숙지한다.**

1. **Function template**

**두 인자의 value를 교환하는myswap function을 template 을 이용하여 작성한다. myswap function template의 두 인자로 references (T& a, T&b)를 입력 받는다. 코드가 완료되면 아래의 예제 코드를 실행시키고, 실제 출력 값과 예제 코드의 출력 결과를 비교한다.**

**int main()**

**{**

**int a = 3, b = 1;**

**myswap(a, b);**

**cout << "a= " << a << " b= " << b << endl;//1, 3**

**float c = 3.1f, d = 1.5f;**

**myswap(c, d);**

**cout << "c= " << c << " d= " << d << endl;//1.5, 3.1**

**return 1;**

**}**

1. **class template**

**function과 마찬가지로 class도 templated 될 수 있다.** Figure 1**과 같이 임의의 형의array를 담을 수 있는 class template, MyContainer를 작성해보자.**

* 1. **이 class template은T형 배열 obj\_arr와 그 배열의 원소 수 n\_elements를 멤버 변수로 가진다.**
  2. **생성자** MyContainer(int n)**는 ‘n’개의 원소를 담을 수 있는 공간을 obj\_arr에 동적으로 할당한다.**
  3. clear() 함수는 할당된 모든 리소스를 해제하고, **n\_elements**를 0으로 reset한다.

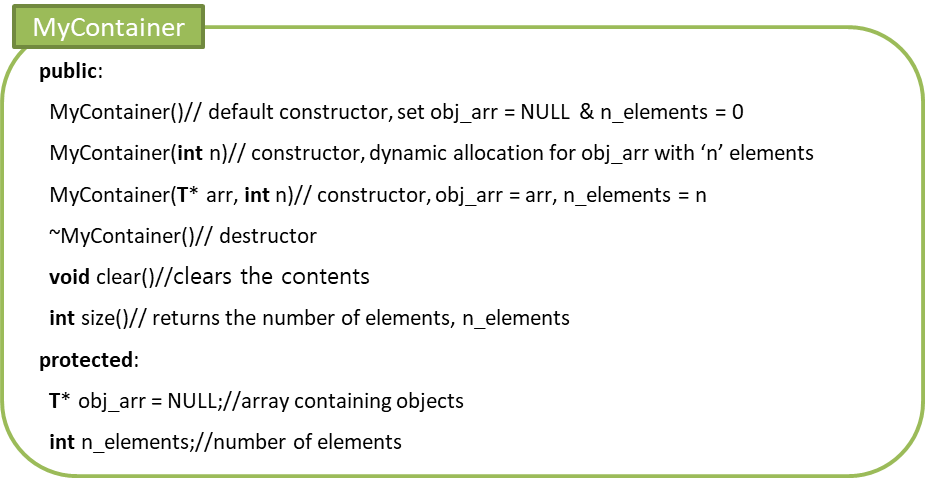


Figure 1. MyContainer class template

1. **class template inheritance**

**class template도 일반 class와 마찬가지로 상속 관계를 구현할 수 있다.** Figure 2**와 같이MyContainer로부터 상속을 받는 class template, MyVector를 작성해보자.**

* 1. **MyVector의 멤버변수, int capacity,는 배열에 담긴 원소의 개수(n\_elements)가 아니라 배열에 담을 수 있는 총 원소의 수를 의미한다. (일반적으로, capacity >= n\_elements)**
  2. **MyVector의 생성자는MyContainer 생성자를 이용하여 생성한다. 이 때, capacity = n\_elements로 함께 초기화 한다.**
  3. **push\_back 멤버 함수는 obj\_arr의 끝(i.e., obj\_arr[n\_elments] )에 새로운 원소를 추가한다. 배열의 끝에 새로운 원소를 추가하기 위해서는 capacity > n\_elements 를 만족해야 한다. 따라서 n\_elements <= capacity인 경우에는, capacity =max(1,2\*capacity) 크기의 배열을 새로 만들고, 새로운 배열의n\_elements+1 번째 자리에 원소를 추가하도록 한다.**
  4. **pop\_back 함수는 obj\_arr의 마지막 원소를 제거한다. 이 때, n\_elements값 도 수정하도록 한다.**

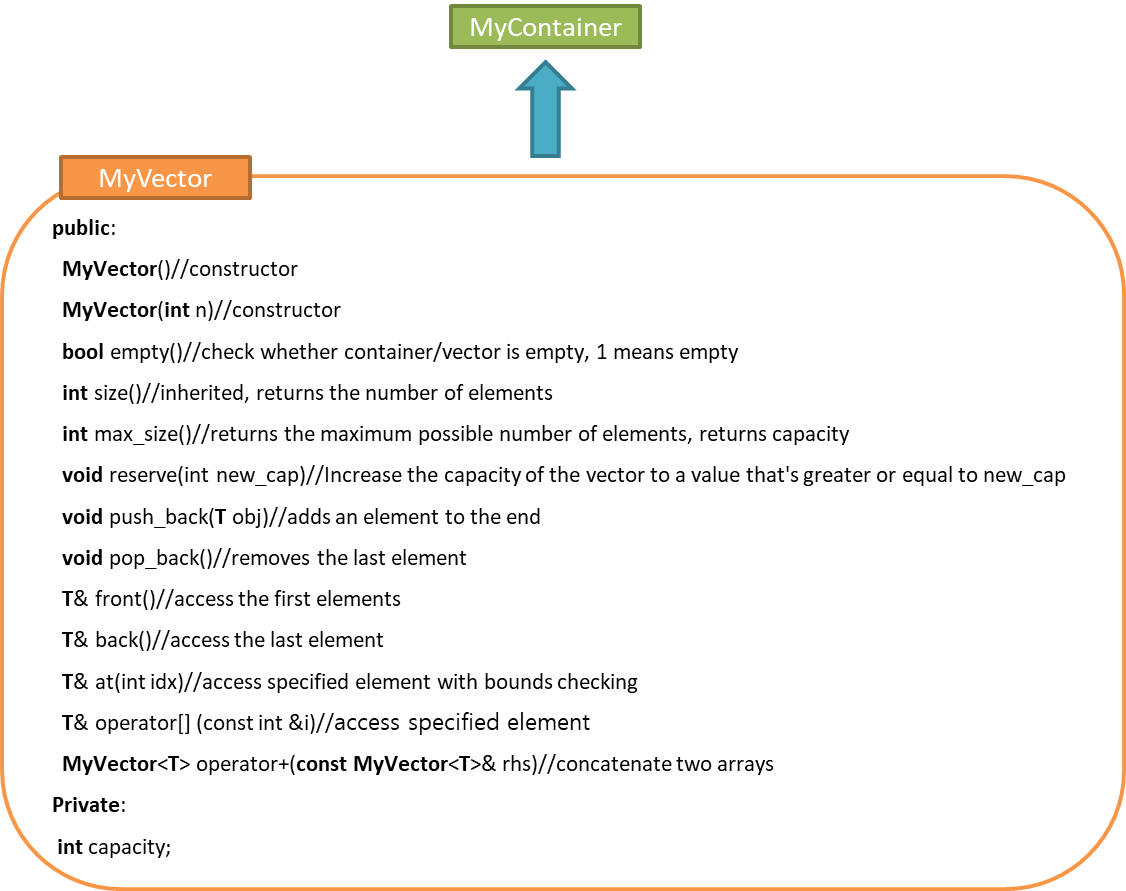


Figure 2. derived class template, MyVector

* 1. **아래 예제 코드를 실행시키고 실제 출력 값이 예제 코드의 출력 값과 동일한지 확인하자.**

**int main(){**

**MyVector<int> myvec1;**

**myvec1.push\_back(0);**

**cout << myvec1.size() << endl;//1**

**cout << myvec1.max\_size() << endl;//1**

**int arr[] = {1, 2, 3, 4, 5};**

**MyVector < int > myvec2(arr, 5);**

**cout << myvec2.size() << endl;//5**

**cout << myvec2.max\_size() << endl;//5**

**myvec2.push\_back(5);**

**cout << myvec2.size() << endl;//6**

**cout << myvec2.max\_size() << endl;//10**

**for(int n = 0; n < myvec2.size(); n++)**

**cout << myvec2[n] << " ";//1,2,3,4,5,5**

**cout << endl;**

**myvec2.back() = 6;**

**for(int n = 0; n < myvec2.size(); n++)**

**cout << myvec2.at(n) << " ";//1,2,3,4,5,6**

**cout << endl;**

**MyVector < int > myvec3 = myvec1 + myvec2;**

**for(int n = 0; n < myvec3.size(); n++)**

**cout << myvec3.at(n) << " ";//0,1,2,3,4,5,6**

**cout << endl;**

**return 1;**

**}**