

# 데이터베이스설계 (ICE4016)

## 과제 #2

---

### ArrayStack

Prof. Wonik Choi

## 과제 2 – ArrayStack 작성

○ 아래의 코드를 바탕으로 ArrayStack을 완성하시오.

### ArrayStack.h

```
template <typename E>
class ArrayStack {
    enum { DEF_CAPACITY = 100 };           // default stack capacity
public:
    ArrayStack(int cap = DEF_CAPACITY);    // constructor from capacity
    int size() const;                      // number of items in the stack
    bool empty() const;                    // is the stack empty?
    const E& top() const throw(StackEmpty); // get the top element
    void push(const E& e) throw(StackFull); // push element onto stack
    void pop() throw(StackEmpty);          // pop the stack
    // ...housekeeping functions omitted
private:                                  // member data
    E* S;                                 // array of stack elements
    int capacity;                          // stack capacity
    int t;                                 // index of the top of the stack
};
```

### ArrayStack.h

```
template <typename E> ArrayStack<E>::ArrayStack(int cap)
: S(new E[cap]), capacity(cap), t(-1) { } // constructor from capacity

template <typename E> int ArrayStack<E>::size() const
{ return (t + 1); } // number of items in the stack

template <typename E> bool ArrayStack<E>::empty() const
{ return (t < 0); } // is the stack empty?

template <typename E> // return top of stack
const E& ArrayStack<E>::top() const throw(StackEmpty) {
    if (empty()) throw StackEmpty("Top of empty stack");
    return S[t];
}

template <typename E> // push element onto the stack
void ArrayStack<E>::push(const E& e) throw(StackFull) {
    if (size() == capacity) throw StackFull("Push to full stack");
    S[++t] = e;
}

template <typename E> // pop the stack
void ArrayStack<E>::pop() throw(StackEmpty) {
    if (empty()) throw StackEmpty("Pop from empty stack");
    --t;
}
```

# 과제 2 – ArrayStack 작성

○ 아래의 코드를 바탕으로 ArrayStack을 완성하시오.

## ArrayStack.h

```
class RuntimeException{
private:
    string errorMsg;
public:
    RuntimeException( const string & err ){
        errorMsg = err;
    }
    string getMessage() const{
        return errorMsg;
    }
};
inline std::ostream& operator<<(std::ostream& out, const RuntimeException& e){
    return out <<e.getMessage();
}
```

```
class StackEmptyException : public RuntimeException{
public:
    StackEmptyException( const string & err ) : RuntimeException(err){
    }
};
```

## ArrayStack.cpp

```
#include <iostream>
#include "ArrayStack.h"
```

```
using namespace std;
```

```
// 주석점수 10점
```

```
// 보고서 점수 20점
```

(1) ArrayStack.h를 수정하여 RuntimeException을 추가하고 이를 상속받는 StackFullException과 StackEmptyException을 구현하라(30점)

```
int main( void )
```

```
{
```

```
    ArrayStack<int> mystack;
```

(2) 아래의 size(), push(), pop(), top() 이 정상 동작함을 보일 것 (20점)

```
    cout << mystack.size() << endl ;
```

```
    mystack.push( 1 );
```

```
    mystack.push( 2 );
```

```
    cout << mystack.top() << endl ;
```

```
    mystack.pop();
```

```
    cout << mystack.top() << endl ;
```

```
    cout << mystack.size() << endl ;
```

(3) 스택이 full인 상태에서 push를 하면 StackFullException이 발생함을 보일 것(20점)

(4) 스택이 empty인 상태에서 pop을 하면 StackEmptyException이 발생함을 보일 것(20점)

```
    return 0;
```

```
}
```

# 주의 사항

---

- 프로그램 작성시 주석은 최대한 많이 작성할 것. 주석 작성도 점수에 포함됨.
- 보고서는 표지(표지는 반드시 l-class의 공지사항에 게시되어있는 서약서표지를 사용할 것), 개요, 구현상 특징, 실행화면 캡처이미지를 반드시 포함하도록 한다. 이때 캡처화면에는 자신의 이름이나 학번이 포함되도록한다.
- Visual Studio 및 기타 IDE를 이용하여 작성할시 프로젝트(또는 솔루션) 이름을 “학번\_Assignment2”로 하고 이 디렉토리를 압축하여 제출할 것. 보고서 파일도 함께 압축하여 업로드하기 바랍니다. 아울러, gcc등 다른 환경으로 작성할시에는 컴파일 방법을 반드시 보고서에 명시하기 바랍니다.

(예) 123456789\_Assignment2 최상위 디렉토리를 **보고서와 함께** 압축한 후,  
123456789\_Assignment2.zip 을 업로드 할 것.

- copy, copied 검출을 엄격하게 수행할 것이며, 부정한 방법으로 과제를 제출한 학생은 심각한 불이익이 가도록 할 예정임.