

# Exam #4



경희대학교  
KYUNG HEE UNIVERSITY

• 다음을 만족하는 프로그램을 작성하시오.

입력:

- 키보드를 이용하여 사용자로부터 긴 문자열을 입력 받는다.

출력:

- 입력된 문자열에서 가장 많이 사용된 단어와 그 횟수를 모두 출력하시오.
- 입력된 문자열에서 가장 적게 사용된 단어와 그 횟수를 모두 출력하시오.

제약사항:

- 문자열을 저장하기 위해 STL의 string 클래스를 이용한다.
- 다음 페이지의 WordElement 클래스와 WordCount 클래스를 구현하고, 이를 이용하여 입력 문자열에서 단어의 수를 센다.

## ● WordElement Class

```
class WordElement
{
public:
    // constructor
    WordElement();
    WordElement(const string& str );
    // copy constructor
    WordElement(const WordElement& wc);

    // destructor
    ~WordElement();

public:
    // operator overloading
    WordElement& operator++( void );           // increment word count
    bool operator==( const string& word );     // compare the strWord to the given string(word)
                                              // returns true if given word and strWord are equal; otherwise false
    bool operator< ( const WordElement& we ); // compare strWord to we.strWord

    // friend << operator
    friend ostream& operator<< (ostream& ostrm, const WordElement & rhs);

    // function to compare by word count
    friend bool CompareCount(const WordElement& first, const WordElement& second);

    // method
    int getCount() const; // get count of a word (returns nCount)

protected:
    string strWord;       // string to store a word
    int    nCount;        // count of strWord
};
```

## • WordCount Class

```
typedef list<WordElement>    LIST_WORDCOUNT;
typedef LIST_WORDCOUNT::iterator    LIST_WORDCOUNTIter;

class WordCount
{
public:
    LIST_WORDCOUNTIter findWord(const string & word);    // Find an element in the m_listWC
    // Find the position of the first occurrence of an element equals to given word in the m_listWC
    // If no such value exists in the m_listWC, returns end position of the m_listWC

    void addWord( const string& word );    // Add an element to the m_listWC
    // If an element equals to given word exists in the m_listWC, increase count of the element;
    // Otherwise, add the element to the m_listWC
    void printMostWords();    // Print the most frequently used word(s)
    void printLeastWords();    // Print the least frequently used word(s)

protected:
    LIST_WORDCOUNT m_listWC;
};
```

## • Main

```
int main ( void )
{
    string strIn;
    WordCount wc;

    cout << "Input string : ";

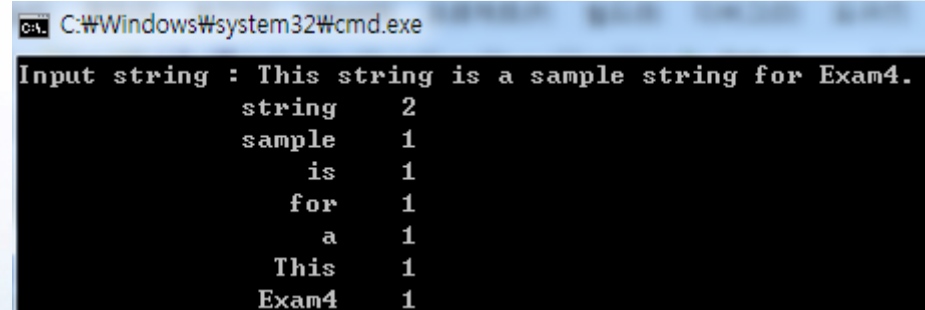
    // Input string

    // build word count list using WordCount and strIn.

    // print the most used word(s)
    wc.printMostWords();
    // print the least used word(s)
    wc.printLeastWords();

    return 0;
}
```

## • Output



```
C:\Windows\system32\cmd.exe
Input string : This string is a sample string for Exam4.
               string      2
               sample      1
                   is      1
                   for      1
                   a        1
                   This      1
                   Exam4     1
```

- 다음 조건을 만족하는 Exception 클래스를 설계, 구현하시오.

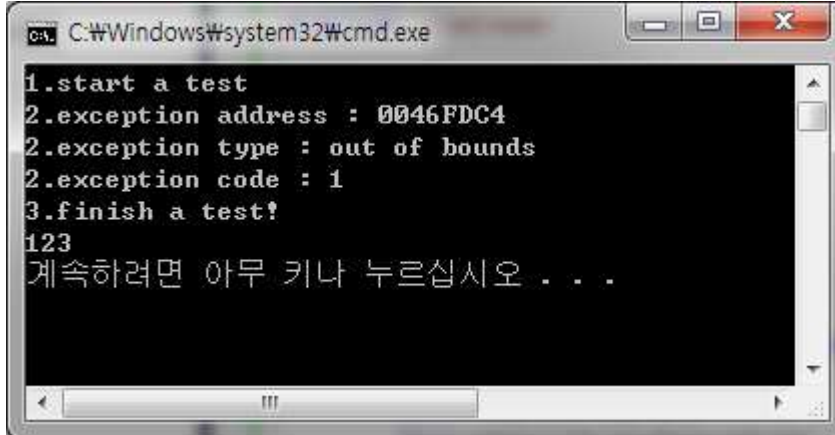
- a) 오류가 발생한 배열의 주소(address)
- b) 오류가 발생한 배열의 인덱스(arrayIndex)
- c) 예외발생 종류(description)
- d) 예외발생 코드(code)

사용 예) main에서 throw로 예외를 던질 때, 다음과 같이 사용됨  
`throw Exception(num,10, "out of bounds",1);`

- 2-1번에서 설계한 Exception 클래스를 이용하여, 다음 코드에 대해 예외를 처리하시오. 11행과 14행 사이를 예외 처리하도록 하고, 2번으로 붙은 메시지를 에러처리 루틴에서 출력하시오.

```
1: #include <cstdlib>
2: #include <new>
3: #include <iostream>
4: using namespace std;
5: class Exception{
6: };
7: int main()
8: {
9:     int num[4];
10:
11:     cout<<"1.start a test\n";
12:     num[10]=123;
13:     throw Exception(num,10, "out of bounds",1);
14:     cout << "3.finish a test!\n";
15:     cout << num[10] << endl;
16:     return 0;
17: }
```

### • Output



```
C:\Windows\system32\cmd.exe
1.start a test
2.exception address : 0046FDC4
2.exception type : out of bounds
2.exception code : 1
3.finish a test!
123
계속하려면 아무 키나 누르십시오 . . .
```



• 다음을 만족하는 프로그램을 작성하시오.

**입력:**

- 키보드를 이용하여 수식을 입력 받는다.

**출력:**

- 검사 결과 괄호의 짝이 맞으면 총 괄호의 개수를 출력하시오.
- 틀리면 'syntax error'를 출력하시오.

**예시:**

입력의 예	출력의 예
$(a+12)/24$	1
$(12+3+4)*99/1.2+(1.1)/99.9+((19*2n)/3)$	4
$((82+4))+42)+(32))$	syntax error

**힌트:**

입력받는 수식에서 처리해야 할 문자는 '('와 ')' 뿐이므로 일반 문자를 만나면 무시하고, '('를 만나면 이를 스택에 저장해 두었다가 ')'를 만나면 스택에서 하나씩 꺼내오는 식으로 문제를 풀 수 있다. 그리고 꺼내올 때마다 사용된 괄호의 개수를 증가시켜 총 개수를 기록해 둔다. 만일 꺼내오려 했는데 스택이 비어있거나, 수식 분석을 마쳤는데 아직 스택에 무언가 남아있다면 잘못된 괄호를 사용했기 때문이다. 수식이 끝난 후 스택이 비어있다면, 괄호의 총 개수를 출력하고, 그렇지 않은 경우엔 syntax error를 출력한다.

## ● Skeleton

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

struct syntax_error{};

int main()
{
    string exp;
    cout<< "Input the expression: " ;
    cin>>exp;

    stack<char> stack;
    size_t cnt(0);
    for(std::string::iterator it=exp.begin(); it != exp.end(); it++)
    {
        switch(*it)
        {

        }

    }

    if(stack.size(>0)
        throw syntax_error();
    else
        cout<<cnt<<endl;
}
```



### • Vehicle 관리를 위한 클래스를 구현한다.

Kyung Hee Motor (KHM) 회사는 bus와 truck의 두 종류의 vehicle을 생산한다. 각 vehicle은 모델 이름과 생산년도를 가진다. 이 정보에 추가로, bus는 color와 총 탑승인원을 속성으로 가진다. Truck은 최대 적재량(truckload)를 추가 속성으로 갖는다. 다음은 각 속성에 대한 데이터 상세이다.

- Model name: string of maximum 20 characters
- Production year: integer
- Bus color: string of maximum 10 characters
- Bus passengers: integer
- Truck load: float

- (a) 다음 페이지의 fn\_a()가 바르게 동작하도록, inheritance와 overriding을 사용하여 Vehicle, Bus, Truck 클래스를 설계하고 구현하시오. (2 points)
- (b) 함수 fn\_b는 vehicle 객체에 대한 포인터를 파라미터로 받아 출력하는 함수이다. Vehicle 객체는 bus 또는 truck 아무거나 될 수 있다. (a)에서 구현한 클래스들은 함수 fn\_b와 사용되기 위해서는 다시 작성되어야 한다. Vehicle\_B, Bus\_B, Truck\_B라는 이름으로 클래스들을 다시 작성하시오. 그리고 void fn\_b(Vehicle\_B const \*)를 작성하시오. (2 point)
- (c) (b)에서 구현한 클래스들에 void printToFile(fstream&) 메소드들을 추가하시오. 이 메소드는 print() 메소드의 출력을 화면 대신에 주어진 file stream에 출력한다. 구현한 메소드들을 사용하여 주어진 vehicle 객체를 지정된 파일에 출력하는 void fn\_c(char const\* filename, Vehicle\_B const\* vehicleObj) 함수를 작성하시오. (2 points)
- (d) Bus 객체들의 배열을 지정된 파일에 기록한 후, 뒤에서 3번째 객체를 파일에서 다시 읽어, 화면에 출력하는 함수 fn\_d를 작성하시오. 파일 입출력은 blocked I/O를 사용한다. (2 points)
- (e) 다음 질문들에 대한 답을 "cout << " 을 사용하여 출력하시오. (1) (a)와 (b)에서 사용한 다형성(polymorphism) 기법들을 설명하시오. (1 point) (2) (b)에서 model name과 bus color에 대해 동적 메모리 할당을 사용하면, 다형성과 관련하여 클래스 구현에서 무엇이 변경되어야 하는 지 설명하시오. (1 point)

## • function main()

```
int main()
{
    cout << "Question (a)" << endl;
    fn_a();

    cout << endl << "Question (b)" << endl;
    Bus_B bigBus("KHM-B1000", 2012, "blue", 45);
    Truck_B smallTruck("KHM-T0500", 2011, 1.5);
    fn_b(&bigBus);
    fn_b(&smallTruck);

    cout << endl << "Question (c)" << endl;
    fn_c("exam04_4.txt", &bigBus);

    cout << endl << "Question (d)" << endl;
    Bus buses[] = {Bus("KHM-B1000", 2012, "blue", 45),
        Bus("KHM-B1050", 2012, "red", 30), Bus("KHM-B1060", 2011, "white", 30),
        Bus("KHM-B3050", 2010, "red", 50), Bus("KHM-B2060", 2010, "black", 60)};
    fn_d("exam04_4.bin", buses, sizeof(buses)/sizeof(Bus));

    cout << endl << "Question (e)" << endl;
    cout << "Answers are here." << endl;

    return 0;
}
```

## • function fn\_a()

```
void fn_a(void)
{
    Bus bigBus("KHM-B1000", 2012, "blue", 45);
    Truck smallTruck("KHM-T0500", 2011, 1.5);

    bigBus.print(); // print bus information
    smallTruck.print(); // print truck information
}
```

## ● 출력 결과

```
C:\Windows\system32\cmd.exe

Question <a>
[BUS]
Model: KHM-B1000
Year: 2012
COLOR: blue
Number of PASSENGERS: 45
[TRUCK]
Model: KHM-T0500
Year: 2011
LOAD: 1.5

Question <b>
[BUS_B]
Model: KHM-B1000
Year: 2012
COLOR: blue
Number of PASSENGERS: 45
[TRUCK_B]
Model: KHM-T0500
Year: 2011
LOAD: 1.5

Question <c>
exam04_4.txt created sucessfully in text mode.

Question <d>
[BUS]
Model: KHM-B1060
Year: 2011
COLOR: white
Number of PASSENGERS: 30
exam04_4.bin created sucessfully in binary mode.

Question <e>
Answers are here.
계속하려면 아무 키나 누르십시오 . . .
```

- 임의의 크기와 임의의 type을 원소로 가질 수 있는 행렬 클래스를 구현하라.

CMatrix클래스는 클래스 템플릿을 사용하여 임의의 type을 원소로 가지는 행렬을 표현할 수 있는 클래스이다.

다음 페이지의 CMatrix 클래스의 선언을 기반으로 클래스를 완성하고 제시한 main함수로 테스트하라.

구현 시 아래 각각의 세부 항목을 순차적으로 만족시켜라. 각각의 세부항목당 부분점수가 있으므로 최대한 만족시켜라.

1. CMatrix를 다음페이지 코드에서 빠진 부분을 채워 넣어서 Template 클래스로 구현하라.
2. 생성자에서는 rows와 cols를 보고 dynamic 하게 data의 메모리를 할당하여 data가 rows x cols크기를 가지는 2차원 배열을 가리키게 만들어야 한다. 그리고 행렬의 내용은 1~100 사이의 랜덤한 수로 채운다.
3. print()로 행렬을 적절한 구조로 출력하게 만들어라.
4. operator=를 구현. 구현 시 양쪽 matrix의 크기가 다르면 에러메시지 출력하고 자신을 리턴하게 하라.
5. 두 matrix를 더하는 operator+를 구현. 양쪽 크기가 다를 시에 에러메시지 출력하고 자신을 리턴하게 하라.
6. float 타입에 대해 print의 specialized member function을 만들어라. 일반 print와 다른점은 마지막 페이지 출력 예 처럼 소수점 출력, 원소간 간격조정 등이다.
7. (추가점수) 수업과 연습시간에 만들었던 fraction 클래스를 추가하고 이 타입의 CMatrix도 모든 기능이 정확히 동작하게 specialized class 를 구현하라.

## • CMatrix class declaration

```
//template class 선언
class CMatrix {
private:
    T **data;        //T type을 가지는 2차원 배열의 포인터.
    int row,col;     //행렬의 가로 세로 크기
public:
    CMatrix( int rows=4, int cols=4);
    //생성자에서는 rows와 cols를 보고 dynamic 하게 data의 메모리를 할당하여
    //data가 rows x cols크기를 가지는 2차원 배열을 가리키게 만들어야 한다.
    //그리고 행렬의 내용은 1~100 사이의 랜덤한 수로 채운다.

    void print();
    //행렬을 프린트 하는 함수

    _____ operator=( const _____ m);
    //어싸인먼트 오퍼레이터 오버로딩

    _____ operator+( const _____ m) const;
    //+오퍼레이터 오버로딩
};
```

## • Main function

```
int main()
{
    srand((unsigned int)time(NULL));

    CMatrix<int> iMat[2];
    cout << "Two integer random matrices\n";
    for(int i = 0;i<2;i++) {
        iMat[i].print();
    }
    CMatrix<int> iMatAnswer = iMat[1] + iMat[0];
    cout << "Sum of two\n";
    iMatAnswer.print();

    CMatrix<float> fMat[2];
    cout << "Two float random matrices\n";
    for(int i = 0;i<2;i++) {
        fMat[i].print();
    }
    CMatrix<float> fMatAnswer = fMat[1] + fMat[0];
    cout << "Sum of two\n";
    fMatAnswer.print();

    cout << "Error handling example\n";
    CMatrix<int> a(3,3), b(4,4);
    a = b + a;

    return 0;
}
```

## ● 출력 결과

Two integer random matrices

4x4Matrix:

```
80  2  44  22
74  42  95  98
56  41  37  26
32  84  83   5
```

4x4Matrix:

```
97  74  80  25
97  37  75  26
75  53  80  83
58  34  76  22
```

Sum of two

4x4Matrix:

```
177  76 124  47
171  79 170 124
131  94 117 109
90 118 159  27
```

Two float random matrices

4x4 Float matrix:

```
17.00  61.00   1.00  83.00
51.00  54.00  86.00  33.00
50.00  73.00  98.00  92.00
92.00  22.00  78.00  42.00
```

4x4 Float matrix:

```
75.00  75.00  82.00  51.00
18.00  60.00   5.00  94.00
71.00  88.00  32.00  47.00
90.00  23.00  67.00  78.00
```

Sum of two

4x4 Float matrix:

```
92.00 136.00  83.00 134.00
69.00 114.00  91.00 127.00
121.00 161.00 130.00 139.00
182.00  45.00 145.00 120.00
```

Error handling example

CMatrix operator +, dimension doesn't match

CMatrix operator =, dimension doesn't match

계속하려면 아무 키나 누르십시오 . . .



- 다음 질문들의 답을 main 함수 뒤 comment 형식으로 삽입하라.

1. 클래스 myClass 타입의 배열 a[100]가 있을 때, a의 값을 찍어보니 16진수로 0013FF10이었고 (a+50)의 값을 찍어보니 00140550 이 나왔다. 그럼 sizeof(myClass)의 값은?
2. Ary[15] + 3 = 5; 와 같은 표현은 컴파일 에러를 발생시킨다. 그 이유를 쓰라.
3. int ary[10]; 과 int\* ary; ary = new int[10]; 의 차이를 설명하라. 각각의 메모리 할당 명칭, 명칭의 이유와 차이, 메모리 할당 시점, 장단점을 모두 설명하여야 한다.
4. template 은 two levels of generalization 이라고 보통 얘기한다. 이의 의미는 무엇인가? Generic class나 function로 부터 concrete class나 function은 어느 시점에 생성되는지에 대한 내용도 답에 포함시켜 설명하라.