

Lab 4

Assignment 1 Supplement

- Concepts of Inheritance and Polymorphism
- Is-a vs Has-a
 - Is-a : Class Person (parent) / Class Student (child)
 - Student IS a Person (not vice versa)
 - Has-a : Class Student / Class UniversityID
 - Student HAS a University ID

Assignment 1 Supplement

- Implementing constructor of Child class
 - You need to specify how to use Parent class constructor as well.
 - ex) Person has member variables age and name; also constructors Person() / Person(int a, string n)
 - Student has additional member variable universityID
 - Student() : Person() { }
 - Student(int a, string n, int univid) : Person(age, name) { }

```
Person::Person() {  
    age = 0;  
    name = "";  
}
```

```
Person::Person(int a, string n) {  
    age = a;  
    name = n;  
}
```

```
Student::Student() : Person() {  
    // age = 0; <- it is already done  
    // name = ""; <- it is already done  
    universityID = 0;  
}
```

```
Student::Student(int a, string n, int univid) : Person(a, n) {  
    //age = a; <- it is already done  
    //name = n; <- it is already done  
    universityID = univid;  
}
```

Assignment 1 Supplement

- Printing out
- You usually use `cout << “~~~” << endl;`
- `endl` : end of line; it automatically inserts new line character “\n”
- Special characters
 - \n: new line character
 - \t: tab

reference: <https://www.dummies.com/programming/cpp/special-character-constants-in-c/>

Vector

- An array-based container that supports a random access iterator.
- Elements are stored consecutively in one memory block.

`template<typename T, typename Allocator = allocator<T>> class vector`

`v.pop_back()` : Remove the last element of v. `v.push_back()`
: Add the element to the end of v

```
#include <iostream>
#include <vector>
using namespace std; int
main(void)
{
    vector<int> v;
    v.push_back(5);
    v.push_back(2);
    v.pop_back();
}
```

Vector

- **vector<int> v;**
 - 비어있는 vector v를 생성합니다.
- **vector<int> v(5);**
 - 기본값(0)으로 초기화 된 5개의 원소를 가지는 vector v를 생성합니다.
- **vector<int> v(5, 2);**
 - 2로 초기화된 5개의 원소를 가지는 vector v를 생성합니다.
- **vector<int> v1(5, 2);**
vector<int> v2(v1);
 - v2는 v1 vector를 복사해서 생성됩니다.
- **vector<int> v1; , vector<int> v2;** 가 있고, 내부에 인자들이 있다고 했을때.

연산자 : "==" , "!=" , "<" , ">" , "<=" , ">=" 로 대소비교 가 가능합니다.

Vector

- **v.assign(5, 2);**
 - 2의 값으로 5개의 원소 할당.
- **v.at(idx);**
 - idx번째 원소를 참조합니다.
 - v[idx] 보다 속도는 느리지만, 범위를 점검하므로 안전합니다.
- **v[idx];**
 - idx 번째 원소를 참조합니다.
 - 범위를 점검하지 않으므로 속도가 v.at(idx)보다 빠릅니다.
- **v.front();**
 - 첫번째 원소를 참조합니다.
- **v.back();**
 - 마지막 원소를 참조합니다.
- **v.clear();**
 - 모든 원소를 제거합니다.
 - 원소만 제거하고 메모리는 남아있습니다.
 - size만 줄어 들고 capacity는 그대로 남아있습니다.
- **v.begin();**
 - 첫번째 원소를 가리킵니다. (iterator와 사용)
- **v.end();**
 - 마지막의 "다음"을 가리킵니다 (iterator와 사용)
- **v.rbegin();**
 - reverse begin을 가리킨다 (거꾸로 해서 첫번째 원소를 가리킵니다)
 - iterator와 사용.
- **v.rend();**
 - reverse end 을 가리킨다 (거꾸로 해서 마지막의 다음을 가리킵니다)
 - iterator와 사용.
- **v.reserve(n);**
 - n개의 원소를 저장할 위치를 예약합니다(미리 동적할당 해놓는다)
- **v.resize(n);**
 - 크기를 n으로 변경한다.
 - 더 커졌을 경우 default값인 0으로 초기화 한다.
- **v.resize(n,3);**
 - 크기를 n으로 변경한다.
 - 더 커졌을 경우 인자의 값을 3으로 초기화한다.

C++ File I/O

- C++ file I/O are easy to implement using the class of ofstream and ifstream in the **fstream** library.
- ofstream: File classes for write operations (derived from ostream)
- ifstream: File classes for read operations (derived from istream)
- **fstream**: File classes for both reads and writes (derived from iostream)

“test.txt” 경로

base.cpp - VSC - Visual Studio Code

lab4 > base.cpp > main()

```
1 #include <fstream>
2 #include <iostream>
3 #include <string>
```

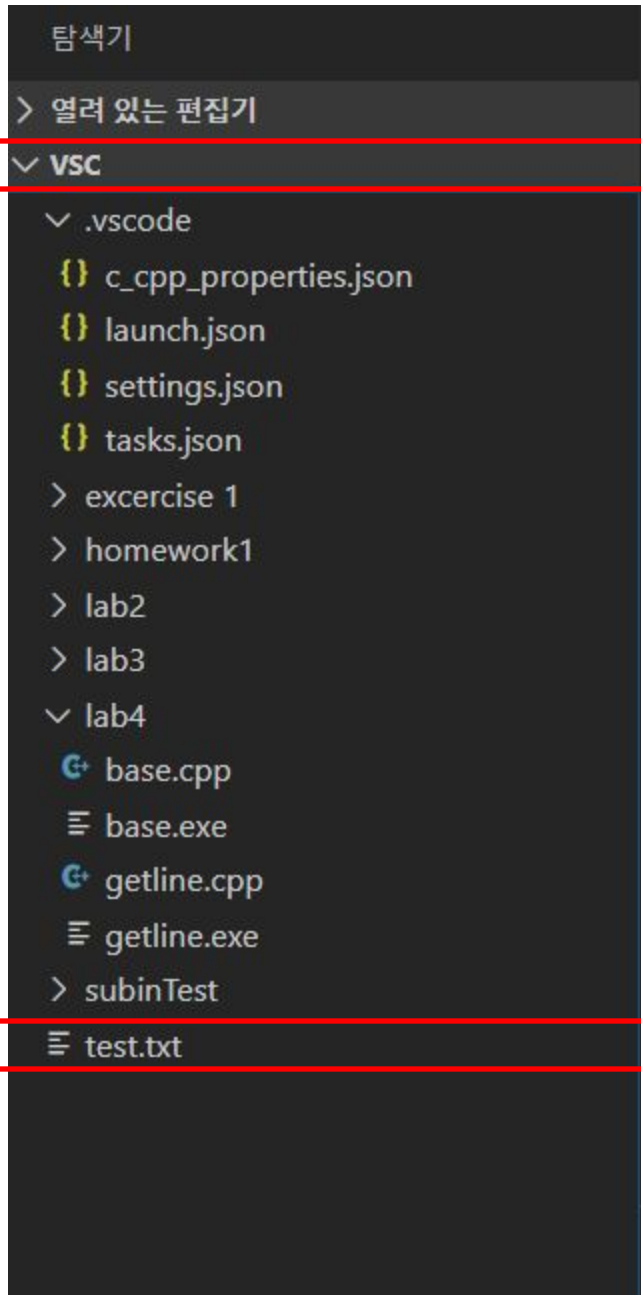
lab4 > base.cpp > main()

```
1 #include <fstream>
2 #include <iostream>
3 #include <string>
4
5 int main() {
6
7     std::ifstream in("C:/Users/SONGMAN/Documents/VSC/lab4/base.cpp/test.txt");
8     std::string s;
9
10    if (in.is_open()) {
11        in >> s;
12        std::cout << "hihi " << s << std::endl;
13    } else {
14        std::cout << "no file" << std::endl;
15    }
16    return 0;
17 }
```

int main() {
 // 파일 읽기 준비
 std::ifstream in("lab4/test.txt");
 std::string s;

(1) 절대 경로

(2) 상대경로



“test.txt” 경로 설정 – 3rd Solution

열려 있는 최상위 디렉토리에서 txt를 만드는 방법

이전 방법과 차이점 : text file의 위치

C++ File I/O - INPUT

```
1  #include <fstream>
2  #include <iostream>
3  #include <string>
4
5  int main() {
6
7      std::ifstream in("test.txt");
8      std::string s;
9
10     if (in.is_open()) {
11         in >> s;
12         std::cout << "input : " << s << std::endl;
13     } else {
14         std::cout << "can't find the file" << std::endl;
15     }
16     return 0;
17 }
```

function open:

open (filename, mode);

Where filename is a string representing the name of the file to be opened, and mode is an optional parameter with a combination of the following flags: (next slide)

- 여기서 in.close() 꼭 안해줘도 됨
 - 안해주면 객체 소멸자에서 해주기 때문

ios::in	Open for input operations.
ios::out	Open for output operations.
ios::binary	Open in binary mode.
ios::app	All output operations are performed at the end of the file, appending the content to the current content of the file.
ios::trunc	If the file is opened for output operations and it already existed, its previous content is deleted and replaced by the new one.

ifstream default parameter

ofstream default parameter

All the flags can be combined using the bitwise operator OR (|).

```
ofstream myfile;
myfile.open ("example.bin", ios::out | ios::app | ios::binary);

//conduct the same opening operation
ofstream myfile ("example.bin", ios::out | ios::app | ios::binary);
```

C++ File I/O – OPEN() & CLOSE()

in 객체를 다른 file로 재사용할 때는 이전에 꺼를 close()
반드시 해줘야 함.

```
1  #include <fstream>
2  #include <iostream>
3  #include <string>
4
5  int main() {
6      // 파일 읽기 준비
7      std::ifstream in("test.txt");
8      std::string s;
9
10     if (in.is_open()) {
11         in >> s;
12         std::cout << "input :" << s << std::endl;
13     } else {
14         std::cout << "can't find the file!" << std::endl;
15     }
16
17     in.close();
18     in.open("other.txt");
19
20     if (in.is_open()) {
21         in >> s;
22         std::cout << "input :: " << s << std::endl;
23     } else {
24         std::cout << "can't find the file!" << std::endl;
25     }
26
27     return 0;
28 }
```



```

1  #include <fstream>
2  #include <iostream>
3  #include <string>
4
5  int main() {
6      // 파일 읽기 준비
7      std::ifstream in("test.txt");
8      std::string s;
9
10     if (in.is_open()) {
11         // 위치 지정자를 파일 끝으로 옮긴다.
12         in.seekg(0, std::ios::end);
13
14         // 그리고 그 위치를 읽는다. (파일의 크기)
15         int size = in.tellg();
16
17         // 그 크기의 문자열을 할당한다.
18         s.resize(size);
19
20         // 위치 지정자를 다시 파일 맨 앞으로 옮긴다.
21         in.seekg(0, std::ios::beg);
22
23         // 파일 전체 내용을 읽어서 문자열에 저장한다.
24         in.read(&s[0], size);
25         std::cout << s << std::endl;
26     } else {
27         std::cout << "can't find the file!" << std::endl;
28     }
29
30     return 0;
31 }

```

File 전체 읽기

seekg(offset, location) :

location 기준으로 offset만큼 이동한 위치를 나타낸다.

tellg() :

커서전까지의 데이터 크기를 알려준다.

ifstream getline()

```
1  #include <fstream>
2  #include <iostream>
3  #include <string>
4
5  int main() {
6      // 파일 읽기 준비
7      std::ifstream in("test.txt");
8      char buf[100];
9
10     if (!in.is_open()) {
11         std::cout << "can't find the file!" << std::endl;
12         return 0;
13     }
14
15     while (in) {
16         in.getline(buf, 100);
17         std::cout << buf << std::endl;
18     }
19
20     return 0;
21 }
```

buffer size를 명시해줘야 함

string:: getline()

```
1 // getline of std::string
2 #include <fstream>
3 #include <iostream>
4 #include <string>
5
6 int main() {
7     // 파일 읽기 준비
8     std::ifstream in("test.txt");
9
10    if (!in.is_open()) {
11        std::cout << "파일을 찾을 수 없습니다!" << std::endl;
12        return 0;
13    }
14
15    std::string s;
16    while (in) {
17        getline(in, s);
18        std::cout << s << std::endl;
19    }
20
21    return 0;
22 }
```

buffer size 없어도 됨

C++ File I/O - OUTPUT

```
1  #include <iostream>
2  #include <fstream>
3  #include <string>
4
5  int main() {
6      // 파일 쓰기 준비
7      std::ofstream out("test.txt");
8
9      std::string s;
10     if (out.is_open()) {
11         out << "helo write world!";
12     }
13
14     return 0;
15 }
```

- File write

ofstream object_name("file_name");

OUTPUT – APPEND

```
1
2  #include <fstream>
3  #include <iostream>
4  #include <string>
5
6  int main() {
7      std::ofstream out("test.txt", std::ios::app);
8
9      out<< "append hello";
10     return 0;
11 }
```

문자열 스트림 (std::stringstream)

```
1  #include <iostream>
2  #include <sstream>
3  #include <string>
4
5  int main() {
6      std::istringstream iss("123");
7      std::ostringstream oss;
8      int x;
9      std::string y;
10
11     iss >> x; // string to integer
12
13     //integer to string
14     oss << x;
15     y=oss.str();
16     //
17
18     std::cout << "input " << x << " " << y << std::endl;
19
20
21     return 0;
22 }
```

C++ File I/O

- `getline(ifstream, string)`
- `getline()`은 개행문자(‘\n’)를 입력의 끝으로 인식하여 정해진 길이만큼 한 줄 전체를 읽어 들인다.

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

int main()

{
    // write File
    ofstream writeFile("test.txt");
    if( writeFile.is_open() ){
        writeFile << "Hello World!\n";
        writeFile << "This is C++ File Contents.\n";
        writeFile.close();
    }

    // read File
    ifstream openFile("test.txt");
    if( openFile.is_open() ){
        string line;
        while(getline(openFile, line)){
            cout << line << endl;
        }
        openFile.close();
    }

    return 0;
}
```

Exercise

- **Pre-requisite knowledge** before Project 1
- String manipulation(Search, Split, Compare, ...)
- **No submit; just for your own benefit**
- Your goal is to show the right result from a text file.
- This week's task is done with cpp.

From a text file with one string line

- Go over to **Visual Studio Code**
- Make a new file titled “fileIOprac.txt”.
- In the text file, copy the following string and save it.

James,1992,10,23,180,male

- Get the string from the file (using ifstream)
- Save that string into a variable which data type is “char []”
- Split the string into the form in the next page.

Sample code (char [] split)

```
#include <iostream>
#include <cstring>
#include <vector>
#include <fstream>
using namespace std;

int main() {
    char str[512];
    char *pch;
    const char* delimiter = ",";
    vector<char*> a;

    string filePath = "fileIOprac.txt";

    ifstream openFile(filePath.data());

    if(openFile.is_open()) {
        openFile.getline(str,512);
        openFile.close();
    }

    printf("Splitting string \"%s\" into tokens:\n", str);
    pch = strtok(str,delimiter);
    while(pch!=NULL) {
        printf("%s\n",pch);
        a.push_back(pch);
        pch = strtok(NULL,delimiter);
    }

    return 0;
}
```

reference: <http://www.cplusplus.com/reference/cstring/strtok/>

output:

```
PS C:\Users\DK\Desktop\cppExample\lab4example> .\fileexample.exe
Splitting string "James,1992,10,23,180,male" into tokens:
James
1992
10
23
180
male
```

Sample code continued

```
#include <iostream>
#include <cstring>
#include <vector>
#include <fstream>
using namespace std;

int main ()
{
    char str[] = "Hello,my-name/is.James";
    char title[] = "RTH";
    int i = 0;

    char * pch;
    const char* delimiter = ",./-";
    vector<char*> a;

    printf ("Splitting string \"%s\" into
tokens:\n",str); pch = strtok (str,delimiter);
    while (pch != NULL)
    {
        printf ("%d: %s\n",++i,pch);

        a.push_back(pch);
        pch = strtok (NULL,delimiter);
    }
    return 0;
}
```

- You can also use multiple delimiters at the same time!

```
PS C:\Users\DK\Desktop\cppExample\lab4example> .\a.exe
Splitting string "Hello,my-name/is.James" into tokens:
1: Hello
2: my
3: name
4: is
5: James
```

You get the idea right?

Please make it your own because you will be using this skill doing your project 1!

Sample code (Split with string data type)

```
#include <iostream>
#include <sstream>
#include <cstring>
#include <vector>
#include <fstream>

using namespace std;

int main()
{
    string filePath = "fileI Oprac.txt"; string
    str;
    ifstream openFile(filePath.data());
    if(openFile.is_open()) {
        getline(openFile, str);
        openFile.close();
    }

    stringstream ss(str);
    vector<string> result;

    while( ss.good() )
    {
        string substr;
        getline( ss, substr, ',' );
        result.push_back( substr);
        cout << substr << endl;
    }
    return 0;
}
```

- You get the same output as the first one.

```
PS C:\Users\DK\Desktop\cppExample\lab4example> .\a.exe
James
1992
10
23
180
male
```