C++ 스터디

4. String 객체와 파일 입출력

D.Com

객체(Object)란 무엇인가?

- 객체지향 프로그래밍(Object-Oriented Programming)은 좀 더 나은 프로그램을 만들기 위한 프로그래밍 방식이다.
- 객체는 상태와 행위로 이루어져 있다. (ex - string의 데이터와 멤버 함수)

내 이름은, 만식이다.
나는 평소 물건을 가지고 있고. 사람들은 나를 통해 물건을 넣고 뺄 수 있지.

- 좌측의 객체를 예시로 하면 '만식'은 객체의 이름(같은 종류여도 개별의 객체들은 각각 다른 이름을 가짐)
- '어떤 물건을 가지고 있는가'가 이 객체의 상태
- '물건을 넣고 빼는 것'이 이 객체의 행위

D.Com

String객체 (구성요소)

- string의 구성요소는 크게 Data와 Functions으로 나눌 수 있다.
- Data (= 멤버 데이터 - 아직은 몰라도 됨)
  - Characters (char 여러 개)

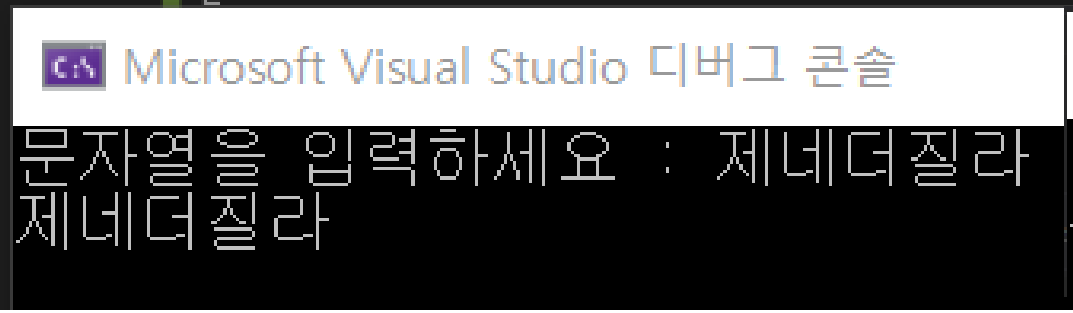
- Functions (= 멤버 함수 혹은 메소드 - 아직 몰라도 됨)
  - 문자열끼리 합치기 (= merge)
  - 문자열에서 char 서치하기
  - 문자열 크기 얻기
  - 문자열에서 모든 char 삭제하기, 등

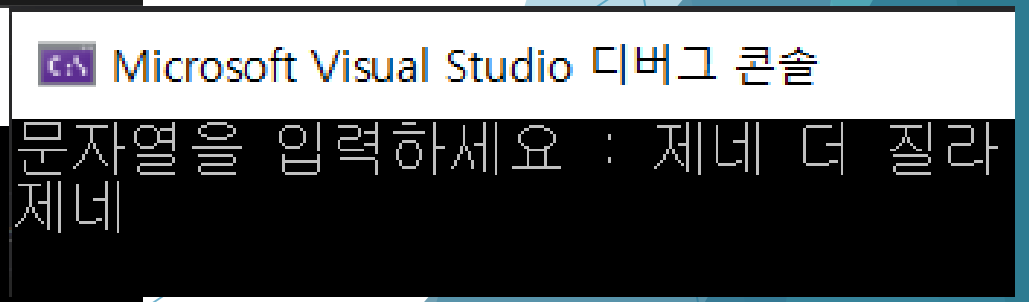
D.Com

string객체 (예제)

- cin 을 이용하여 문자열을 입력할 때에는 공백 단위나 엔터(줄바꿈)을 기준으로 입력을 받기에 띄어쓰기가 있는 문장은 공백을 기준으로 잘려서 들어가게 된다.

```
1   #include <iostream>
2   #include <string> // 문자열을 사용하기 위해서는
3   // string 헤더를 인클루드 해줘야한다.
4   using namespace std;
5
6   int main() {
7       string str;
8       cout << "문자열을 입력하세요 : ";
9       cin >> str;
10      cout << str << endl;
11  }
```





D.Com

string객체 (멤버 함수 or 메소드)

- operator[n] | 문자열의 n번째 요소 반환
- operator= | 하나의 문자열을 다른 문자열에 부여
- operator+= | 문자열 객체의 끝에 한 문자나 문자열을 추가함
- at | 오류 체크 기능을 포함한 문자열의 n번째 요소 반환
- length | 문자열의 길이 반환
- size | 문자열의 길이 반환 (length와 동일)
- find | 검색할 문자열을 받아 그 문자열의 위치 반환
- substr | 문자열 일부를 떼서 새로운 문자열로 반환
- empty | 문자열이 비었는지에 대한 여부, 비었으면 true
- clear | 문자열에서 모든 글자를 지움

D.Com

string객체 (멤버 함수 or 메소드)

- 더 많은 메소드들은 http://www.cplusplus.com/reference/string/string/ 참조
- 한글은 한글자에 2바이트라 [] 오퍼레이터나 at 함수로 접근할 수 없다.

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

int main() {
    string word = "₩'YUMDDA₩'";
    cout << word.length() << endl;
    if (word.empty())
        cout << "empty" << endl;
    else
        cout << "not empty" << endl;
    word.clear();

    if (word.empty())
        cout << "empty" << endl;
    else
        cout << "not empty" << endl;
    word = "돈 call";
    cout << word << endl;
    word += " me";
    cout << word.length() << endl; //한글은 1글자 2바이트!!
    cout << word << endl;
    cout << word[1] << endl; // 한글은 2바이트라 암것두 안나옴
    cout << word[word.length() - 1] << endl;
    cout << word.substr(3, 4) << endl;
    // (첫번째 인덱스, 길이)
}
```

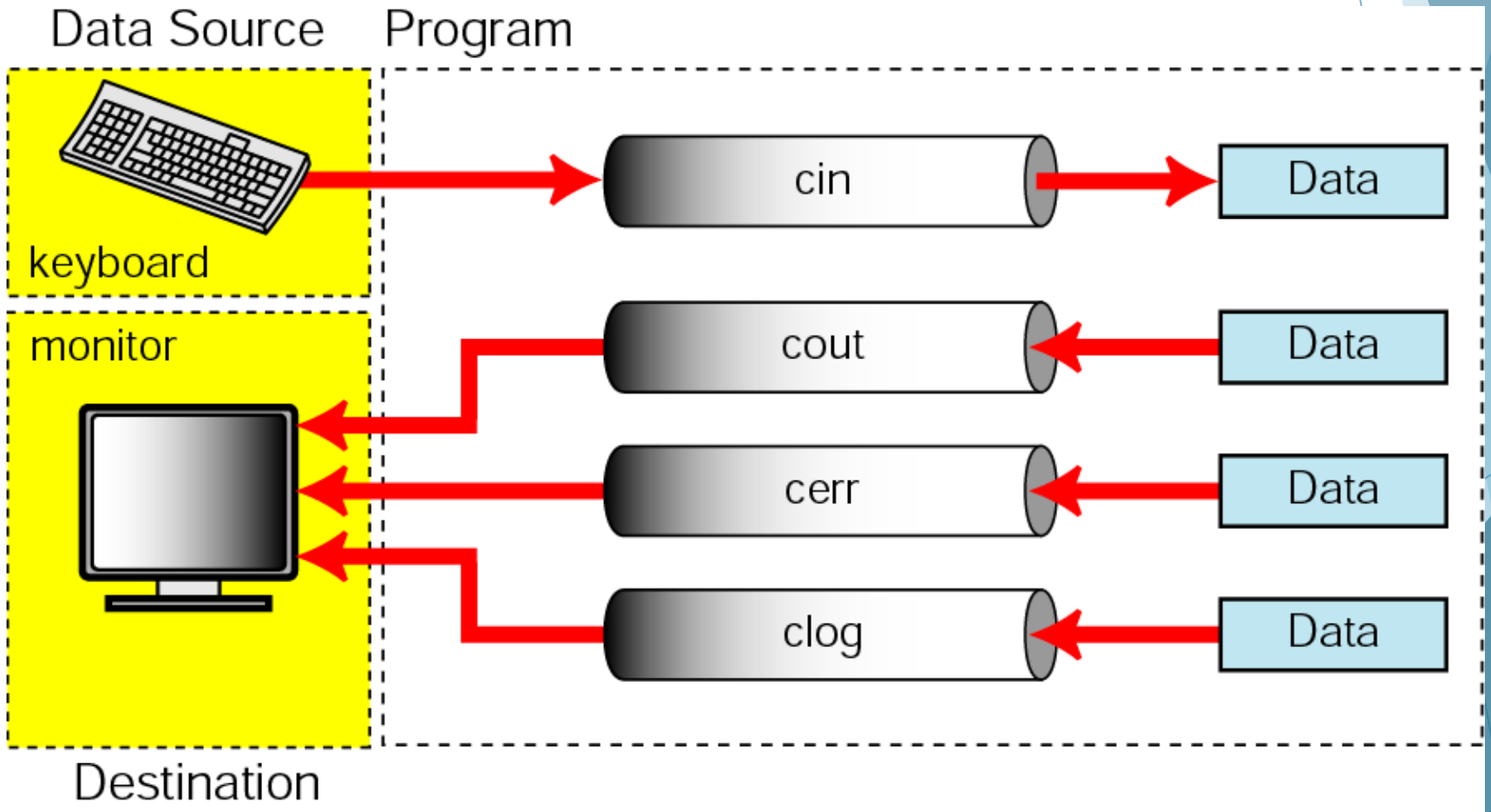
```
Microsoft Visual Studio 디버그 콘솔
8
not empty
empty
돈 call
10
돈 call me

e
call

C:₩Users₩bluej₩Desktop₩방학프
종료되었습니다.
이 창을 닫으려면 아무 키나 누
```

인덱스 = 위치

D.Com

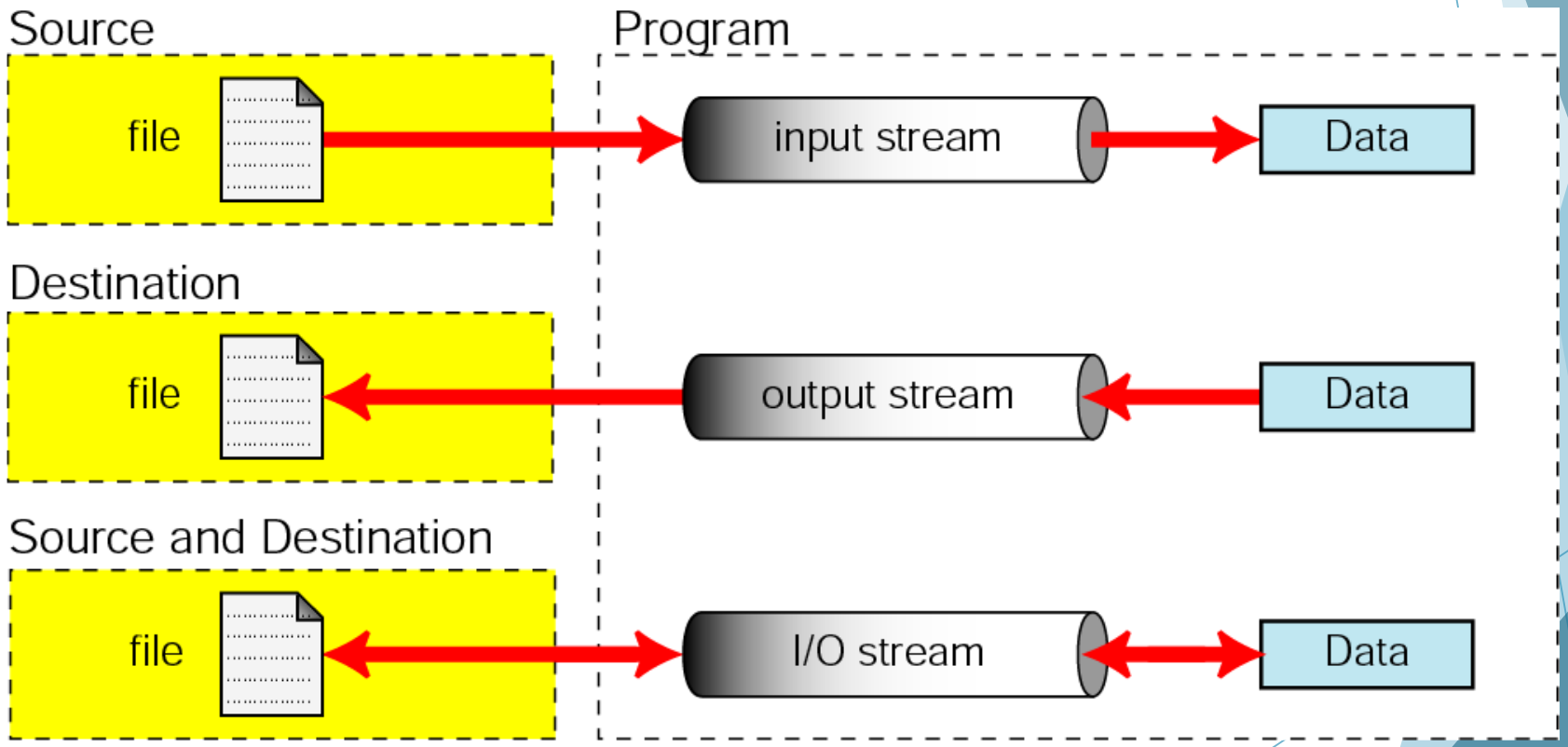
파일 입출력 (Console Stream)

- Console Stream은 콘솔(그 검은창 ㅇㅇ)에 데이터를 입력 받고, 출력하기 위해 사용하는 데이터가 오고 가는 strea이다. 우리는 cout, cin 객체를 이용하여 콘솔에서 입출력을 받아 왔다.



파일 입출력 (File Stream)

- File Stream은 마찬가지로 파일에 데이터를 입력 받고, 출력하기 위해 사용하는, 데이터가 오고가는 strea이다.



D.Com

파일 입출력 (input stream, output stream)

- iostream과 fstream 객체를 정리하자면 다음과 같다.
- fstream을 이용하면 파일 입출력을 동시에 할 수 있으나 보통은 구분하여 사용한다.

목적지	입력 stream	출력 stream
콘솔	cin	cout
파일	ifstream	ofstream

stream이 무엇인지 모르겠다면 데이터가 흐르는 터널로 생각하자.
cin은 키보드와 데이터 사이의 터널인 것이다.
마찬가지로 ofstream은 파일과 데이터 사이의 터널이다.

D.Com

파일 입출력 (File Stream & 기본 멤버 함수)

- C++의 File Stream
  - ifstrem (파일을 읽어 오기 위해 사용)
  - ofstream (파일을 수정하기 위해 사용)
  - fstream (읽고 수정 둘 다)
- File Stream 연결 하기 및 끊기

```
1   #include <iostream>
2   #include <fstream> //파일 입출력을 위해 필요한 헤더
3   using namespace std;
4
5   int main() {
6       ifstream fin; // 스트림종류 스트림객체이름;
7       //cin과의 통일성을 위해 fin으로 정해준거임(딴거 가능)
8
9       fin.open("파일이름.확장자"); //파일 열기
10
11      fin.close(); //다 사용하면 반드시 닫아줘야함
12  }
```

D.Com

파일 입출력 (>>, << operator)

- #include <fstream>을 통해 File Stream 관련 라이브러리를 include 해 ifstream 객체를 사용해 문자열과 정수를 읽는 모습이다. ifstream의 사용법은 cin과 유사하다.

```
1   #include <iostream>
2   #include <fstream>
3   #include <string>
4   using namespace std;
5
6   int main() {
7       ifstream fin; // ifstream 객체 선언
8       fin.open("이름.확장자");
9       string str;
10      int num;
11      fin >> str >> num;
12      cout << str << num << endl;
13      fin.close(); // 파일 닫기(충돌 방지를 위해 필수)
14  }
```

D.Com

파일 입출력 ((>>, << operator)

- 반대로도 가능하다. ofstream 객체를 이용해 문자열을 쓰는 모습이다. ofstream의 사용법은 cout과 유사하다.

```
1  #include <iostream>
2  #include <fstream>
3  #include <string>
4  using namespace std;
5
6  int main() {
7      ofstream fout;
8      string buf = "롤토체스꿀잼";
9      fout.open("output.txt");
10     fout << buf;
11     fout.close();
12 }
```

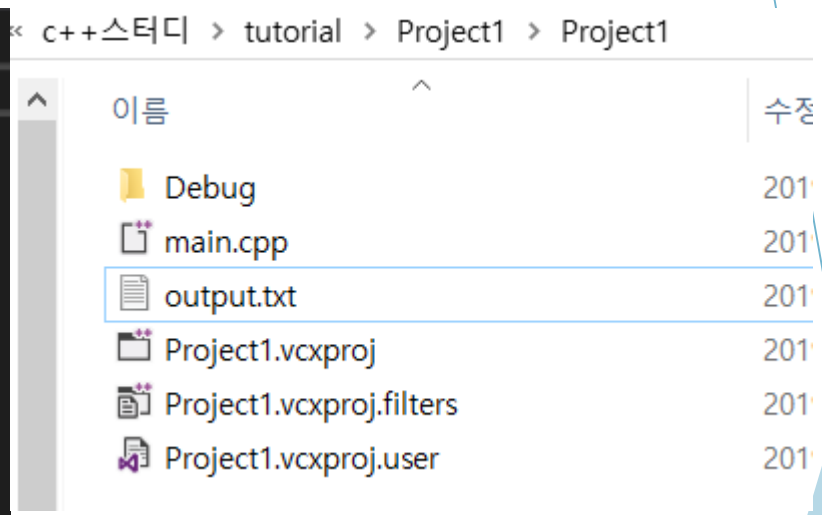
output.txt - 메모장

파일(F) 편집(E) 서식(O)

롤토체스꿀잼

Microsoft Vi

C:₩Users₩blu
₩Project1₩De
로 인해 종료
이 창을 닫으



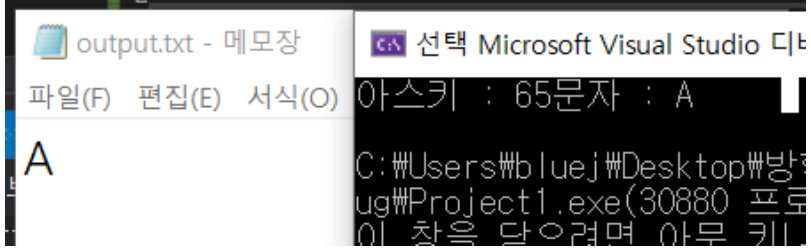
자신의 소스코드가 있는 폴더에 생기며, 파일이 없다면 새로 생성하며 존재할 경우 기본으론 덮어씌운다.

D.Com

파일 입출력 (get, put)

- get 혹은 put 함수를 사용해 문자 하나를 읽고 쓰기도 가능하다. get은 읽은 char의 아스키 코드 (int)를 반환한다. put은 아스키 코드를 문자로 바꿔 넣는다.

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

int main() {
    ifstream fin;
    ofstream fout;
    int buf; // 아스키코드는 int이니

    fout.open("output.txt");
    fout.put(65); // 아스키코드->char 변환해 감
    fout.close();

    fin.open("output.txt");
    buf = fin.get();
    cout << "아스키 : " << buf <<
    "문자 : " << char(buf) << endl; // 길어서 ㅇㅇ
    fin.close();
}
```



파일 입출력 (getline)

- string 라이브러리의 getline 함수를 이용, cin 대신 파일 스트림을 넣어서 파일을 한줄씩 읽을 수 있다. getline은 읽을 줄이 없을 시 false를 반환한다. is_open 함수는 파일 스트림이 제대로 열렸을 때 true를 반환한다.
- 더 자세한 getline 함수에 대해서는 https://modoocode.com/236 참고

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

int main() {
    string line;
    ifstream fin;
    fin.open("input.txt");
    if (fin.is_open()) {
        while (getline(fin, line)) {
            cout << line << endl;
        }
    }
    fin.close();
}
```

```
선택 Microsoft Visual Studio 디버그 콘솔
오늘 새벽도 배고프다.
불닭볶음면
팔도비빔면
와규동
부타동
부타센세 가고싶다 ㅠㅠㅠㅠ
```

D.Com

Data Format

- ios_base(iostream과 fstream의 상위 라이브러리의 멤버함수를 사용해서 데이터를 formatting 가능하다. (cout, cin에도 똑같이 사용가능 하다는 말)
- 다음의 출력이나 입력에만 적용된다!

```
1  #include <fstream>
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      ofstream fsTemp;
7      fsTemp.open("temp.txt");
8      int a = 123;
9      double b = 12.12345678;
10     fsTemp.width(15);
11     fsTemp << a << endl;
12     fsTemp.width(15);
13     fsTemp.precision(10);
14     fsTemp << b << endl;
15     fsTemp.close();
16     return 0;
17 }
```

- width() : 데이터 출력되어 보여질 넓이를 정한다. (데이터가 width보다 작을 경우 공백으로 채움)
- fill() : 설정한 공백이 데이터 넓이보다 클 경우 채울 문자를 정한다. (공백대신에 무엇으로 채울지 정하는 것)
- precision() : 소수점 이후에 올 숫자의 수를 결정한다.

D.Com

Stream Flags

- 마찬가지로 ios_base의 서식 플래그를 사용해 서식을 설정 가능하다.
- setf()를 통해 설정, unsetf()를 통해 해제할 수 있다. 한번 설정하면 그 아래에선 계속 적용된다.
- 사용 형태는 **입출력객체.setf(ios::플래그)**이다. 기능은 아래 사진을 참조.

Flag	Meaning
boolalpha	Boolean values can be input/output using the words "true" and "false".
dec	Numeric values are displayed in decimal.
fixed	Display floating point values using normal notation (as opposed to scientific).
hex	Numeric values are displayed in hexidecimal.
internal	If a numeric value is padded to fill a field, spaces are inserted between the sign and base character.
left	Output is left justified.
oct	Numeric values are displayed in octal.
right	Output is right justified.
scientific	Display floating point values using scientific notation.
showbase	Display the base of all numeric values.
showpoint	Display a decimal and extra zeros, even when not needed.
showpos	Display a leading plus sign before positive numeric values.
skipws	Discard whitespace characters (spaces, tabs, newlines) when reading from a stream.
unitbuf	Flush the buffer after each insertion.
uppercase	Display the "e" of scientific notation and the "x" of hexidecimal notation as capital letters.

Stream Flags (예시)

- 어려우면 예시를 해보자~

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

int main() {
    char aChar;
    cout << "<스페이스키 그리고 z>를 입력해주세요 : ";
    cin >> aChar; //공백이 무시되고 z만 들어감
    cout << "char 입력은 :" << aChar << endl;
    cin.unsetf(ios::skipws); //공백무시 해제
    cin >> aChar; //없으면 실행 안됨.

    cout << "<스페이스키 그리고 z>를 입력해주세요 : ";
    cin >> aChar;  //공백이 입력으로 들어감
    cout << "char 입력은 :" << aChar << endl;
}
```

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

int main() {
    cout << "기본 정렬 : ";
    cout.width(10);
    cout << 12345 << endl;

    cout << "좌측 정렬 : ";
    cout.setf(ios::left, ios::adjustfield);
    cout.width(10);
    cout << 12345 << endl;

    cout << "우측 정렬 : ";
    cout.setf(ios::right, ios::adjustfield);
    cout.width(10);
    cout << 12345 << endl;

    cout << "우측&fill : ";
    cout.fill('-'); // 빈공간 채우기 반드시 char
    cout.width(10);
    cout << 12345 << endl;
}
```

D.Com

파일 입출력 (예시 [1/5])

```
1  /* Create a grades file for transmission to Registrar.
2  Written by:
3  Date:
4  */
5  #include <iostream>
6  #include <fstream>
7  #include <iomanip>
8  #include <cstdlib>
9  //#include <StdAfx.h>
10 using namespace std;
11
12 bool getStu (ifstream& stuFile, int& stuID, int& exam1, int& exam2, int& final);
13 void writeStu (ofstream& gradesFile, int stuID, int avrg, char grade);
14 void calcGrade (int exam1, int exam2, int final, int& avrg, char& grade);
15
16 int main ()
17 {
18     ifstream stuFile;
19     cout << "Begin student grades\n";
20     stuFile.open ("ch7STUFL.DAT");
21     if (!stuFile)
22     {
23         cerr << "\aError opening student file\n";
24         exit (100);
25     } // if open file
26
```

파일 입출력 (예시 [2/5])

```
27      ofstream gradesFile;
28      gradesFile.open ("ch7STUGR.DAT");
29      if (!gradesFile)
30      {
31          cerr << "\aError opening grades file\n";
32          exit (102);
33      } // if open fail
34
35      int stuID;
36      int exam1;
37      int exam2;
38      int final;
39      int avrg;
40      char grade;
41
42      while (getStu (stuFile, stuID, exam1, exam2, final))
43      {
44          calcGrade (exam1, exam2, final, avrg, grade);
45          writeStu (gradesFile, stuID, avrg, grade);
46      } // while
47
48      stuFile.close ();
49      gradesFile.close();
50      cout << "End student grades\n";
51      return 0;
52  } // main
```

파일 입출력 (예시 [3/5])

```
53 /* ==============================getStu==============================
54 Reads data from student file.
55
56 Pre stuFile is an open file stuID, exam1, exam2, final-ref's to int
57 Post reads student ID and exam scores into parameter references
58 Return true if data read -- false if end of file
59 */
60 bool getStu (ifstream& stuFile, int& stuID, int& exam1, int& exam2, int& final)
61 {
62     //!stuFile is either eof or bad file
63     stuFile >> stuID >> exam1 >> exam2 >> final;
64     if (!stuFile)
65         //Read problem or at end of file
66         return false;
67     return true;
68 } // getStu
69
70 /* ==============================clacGrade==============================
71 Determine student grade based on absolute scale.
72 Pre exam1, exam2, and final contain acores avrg and grade are references to variables
73 Post Average and grade copied to references
74 */
75 void calcGrade (int exam1, int exam2, int final, int& avrg, char& grade)
76 {
77     avrg = (exam1 + exam2 +final) / 3;
78     if (avrg >= 90)
```

D.Com

파일 입출력 (예시 [4/5])

```
79          grade = 'A';
80      else if (avrg >= 80)
81          grade = 'B';
82      else if (avrg >= 70)
83          grade = 'C';
84      else if (avrg >= 60)
85          grade = 'D';
86      else
87          grade = 'F';
88      return;
89  } // calcGrade
90  /* ==============================writeStu==============================
91  Writes student grade data to output fiel.
92  Pre gradesFiel is an open file stuID, avrg, and grade have values to write
93  Post Data written to file
94  */
95  void writeStu (ofstream& gradesFile, int stuID, int avrg, char grade)
96  {
97      gradesFile.fill ('0');
98      gradesFile << setw (4) << stuID;
99      gradesFile.fill (' ');
100     gradesFile << setw (3) << avrg;
101     gradesFile << ' ' << grade << endl;
102     return;
103 } // writeStu
```

파일 입출력 (예시 [5/5])

```
/*  Results:
Begin student grades
End student grades

Input--------
0090 90 90 90
0089 88 90 89
0081 80 82 81
0079 79 79 79
0070 70 70 70
0069 69 69 69
0060 60 60 60
0059 59 59 59

Output----
0090 90 A
0089 89 B
0081 81 B
0079 79 C
0070 70 C
0069 69 D
0060 60 D
0059 59 F
*/
```

과제 4

- 랩 4번을 푸세요~

D.Com