

SWING 32기 스터디

[ C++ ]

1주차: C++ 개념, 입출력, 연산



소프트웨어를 만들기 위한 언어  
기계어, 어셈블리어, 고급언어 등이 있음

그중에서 우리가 배울 언어는 고급언어 중에 **C++**

## C언어

---

고급 언어 중 한 언어

대부분의 운영체제 커널은 C언어 이용

시스템 프로그래밍과 응용 프로그래밍에서 많이 쓰임

---

## C++

---

C언어의 문법 체계 계승하고 몇몇 기능을 추가한 언어

---

# 절차 지향과 객체 지향

절차 지향	객체 지향
C	C++, JAVA, Python 등..
순차적 명령어 처리	객체로 선언 후 객체 간의 관계를 처리
-	중복되는 프로세스를 줄이는 등 하드웨어 처리량을 줄임

## 캡슐화

데이터를 객체 안에 감싸서  
외부 접근으로부터 보호  
→ 이러한 특성을 캡슐에 비유

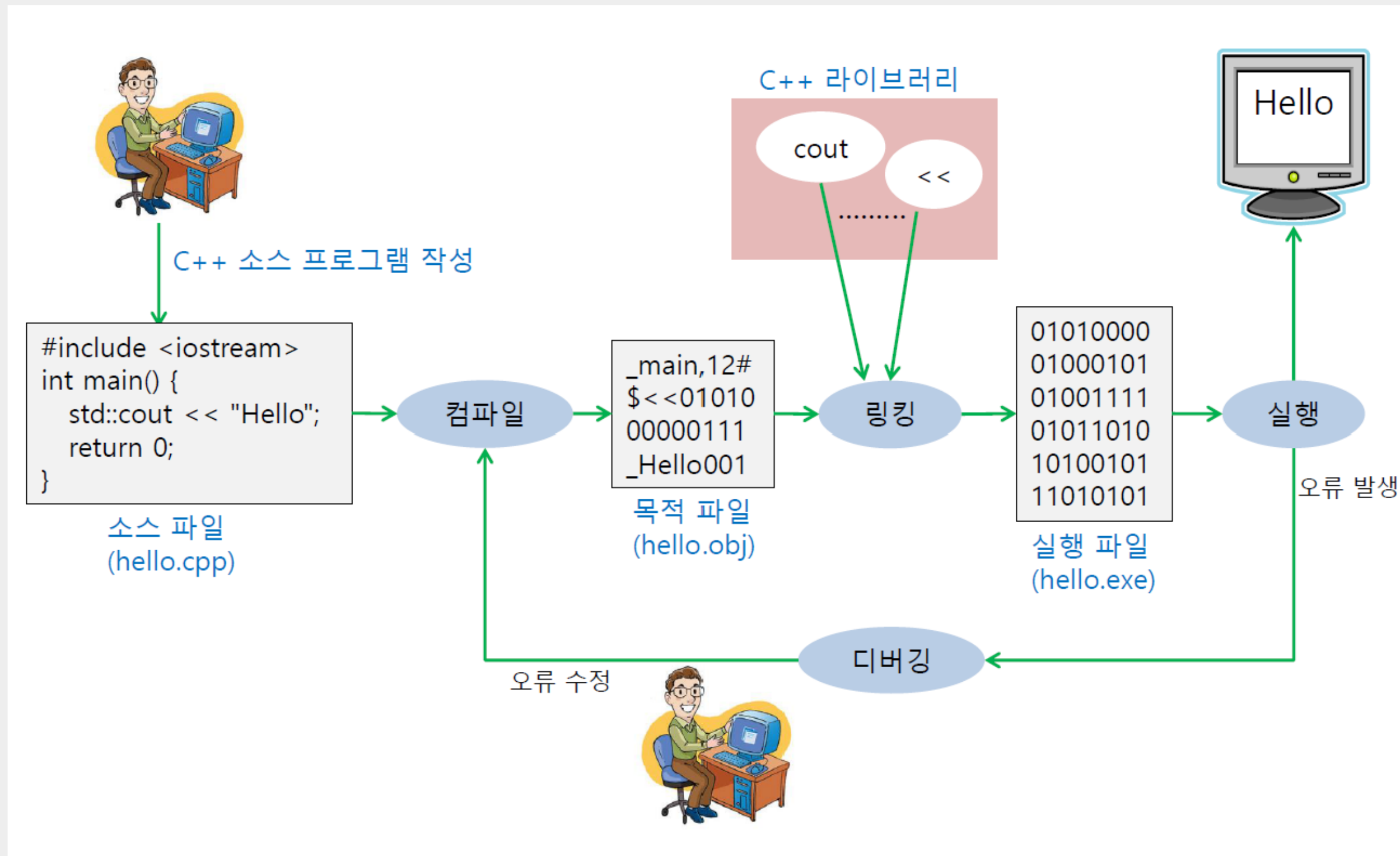
## 객체 지향 3속성

## 상속

상하위 객체 간  
멤버 상수, 변수 상속 가능

## 다형성

중복되는 꼴을  
하나의 기능으로 정의 가능



전처리문

```
#include  
#define
```

사용자 정의 함수

2주차 내용..

메인 함수

```
int main() {  
...  
return 0;  
}
```

```
int main() {  
    //코드 작성  
    return 0;  
}
```

## main 함수

프로그램이 시작되는 지점  
프로그램에서 가장 먼저 실행

**한 개 존재**  
적어서도 많아서도 안됨!

main 함수 종료 -> 프로그램 종료



# return 0

09

```
int main() {  
    //코드 작성  
    return 0;  
}
```

## return문

호출 당한 함수에서  
호출한 함수로 값을 반환하면서  
호출 당한 함수는 종료

main 함수가 반환 -> 프로그램 종료

main 함수는 return문이 없어도  
자동으로 종료 처리(return 0)

```
cppstudy (main.cpp)  
#include <iostream> //컴파일러가 제공하는 헤더 파일  
#include "swing.h" //소스 파일 내에 있는 헤더 파일 (4주차에서 자세히..)  
  
#define X 3.14
```

## 전처리문

코드 컴파일 전 처리해야 할 작업

`#include <헤더 파일>`  
해당하는 파일 내용을  
현재 파일에 포함시킴

`#define`  
매크로나 상수 처리할 때 사용

# <iostream> 라이브러리

---

11

01

`cout <<`

---

C++ 표준 입출력 라이브러리

02

`cin >>`

---

C언어의 printf, scanf와 비슷한 역할 수행

# <iostream> 라이브러리

12

01

cout <<

C++에서 출력을 위한 명령어

<< 라는 삽입 연산자를 이용해 변수(문자열) 출력

```
#include <iostream>
//전처리에 라이브러리를 사용한다는 걸 명시해줘야 함!

int main() {
    int a = 10;
    float b = 0.1;
    char c = 'a';
    char d = 65; //ascii 코드

    std::cout << "반갑습니다\n";
    std::cout << "int형도 " << a << ".. float형도 " << b << "..";
    std::cout << "char형도 " << c << ", " << d << " 모두 출력 가능합니다.";
    return 0;
}
```

Microsoft Visual Studio 디버그

반갑습니다  
int형도 10.. float형도 0.1..char형도 a, A 모두 출력 가능합니다.

# <iostream> 라이브러리

13

01

cout <<

C++에서 출력을 위한 명령어

<< 라는 삽입 연산자를 이용해 변수(문자열) 출력

줄바꿈 할 때, '\n' 대신 **std::endl** 사용 가능

```
#include <iostream>
//전처리에 라이브러리를 사용한다는 걸 명시해줘야 함!

int main() {
    int a = 10;
    float b = 0.1;
    char c = 'a';
    char d = 65; //ascii 코드

    std::cout << "반갑습니다" << std::endl;
    std::cout << "int형도 " << a << ".. float형도 " << b << "..";
    std::cout << "char형도 " << c << ", " << d << " 모두 출력 가능합니다.";
    return 0;
}
```

Microsoft Visual Studio 디버그 ×

+ ▾

반갑습니다  
int형도 10.. float형도 0.1..char형도 a, A 모두 출력 가능합니다.

# <iostream> 라이브러리

14

02

cin >>

C++에서 입력을 위한 명령어

>> 라는 추출 연산자를 이용하여 값을 입력해줌

```
#include <iostream>
//전처리기에 라이브러리를 사용한다는 걸 명시해줘야 함!

int main() {
    int a;

    std::cout << "C++에서 입력 받기: ";
    std::cin >> a;
    std::cout << a;

    return 0;
}
```



Microsoft Visual Studio 디버그



C++에서 입력 받기: 6  
6

# <iostream> 라이브러리

15

02

cin >>

C++에서 입력을 위한 명령어

>> 라는 추출 연산자를 이용하여 값을 입력해줌

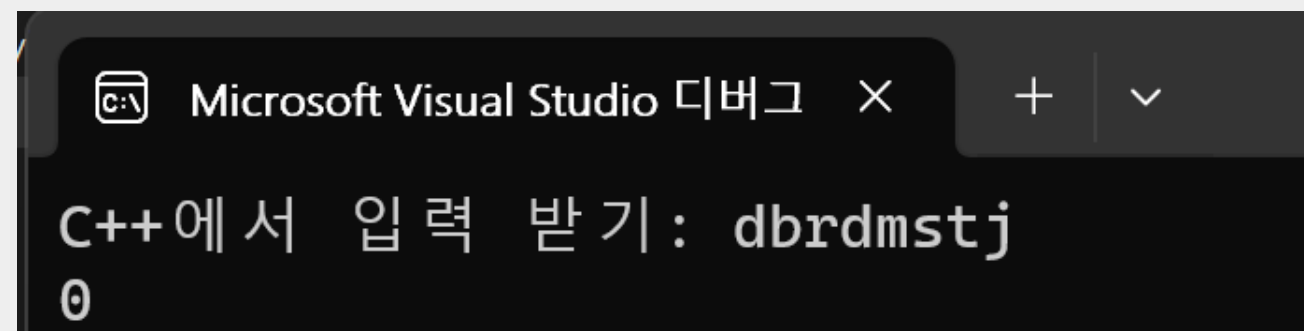
자료형 범위에서만 입력 받을 수 있음

```
#include <iostream>
//전처리기에 라이브러리를 사용한다는 걸 명시해줘야 함!

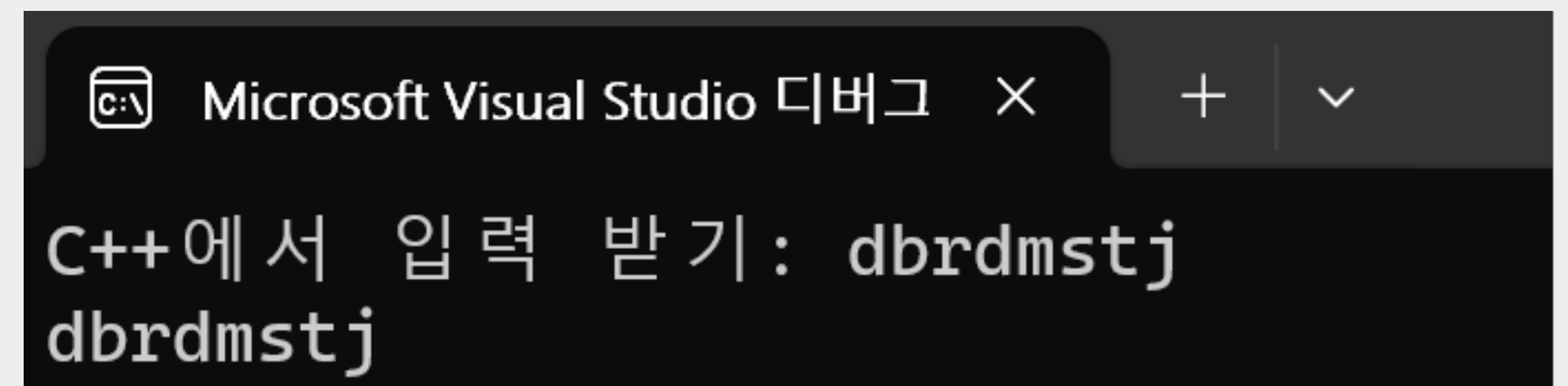
int main() {
    char name[100]; //전 페이지와 다르게 char형 변수 선언

    std::cout << "C++에서 입력 받기: ";
    std::cin >> name;
    std::cout << name;

    return 0;
}
```



\*이전 코드처럼 int형 변수에 입력해주면 0이 뜬다



# <iostream> 라이브러리

16

02

cin >>

C++에서 입력을 위한 명령어

>> 라는 추출 연산자를 이용하여 값을 입력해줌

자료형 범위에서만 입력 받을 수 있음

공백 문자를 받을 때까지 입력 가능

```
#include <iostream>
//전처리기에 라이브러리를 사용한다는 걸 명시해줘야 함!

int main() {
    char name[100]; //전 페이지와 다르게 char형 변수 선언

    std::cout << "C++에서 입력 받기: ";
    std::cin >> name;
    std::cout << name;

    return 0;
}
```

Microsoft Visual Studio 디버그

C++에서 입력 받기: 31기 육은서  
31기



# C++에서 문자열 표현

17

## C++에서 문자열 표현

### C언어 계승

문자열을  
문자들의 집합으로 인식  
문자열을 상수 취급

### string

문자열을 위한 자료형  
문자열을 변수 취급

```
int main() {  
    char a = 'a'; //작은 따옴표(')로 문자 표현  
    char abcd[5] = "abcd"; //큰 따옴표("")로 문자열 표현  
    //abcd는 네 글자이지만 널바이트를 생각해서 범위를 지정해줘야 함  
  
    return 0;  
}
```

char형 -> 문자를 위한 자료형  
string형 -> 문자열을 위한 자료형

# C++에서 문자열 표현

18

## C++에서 문자열 표현

### C언어 계승

문자열을  
문자들의 집합으로 인식  
문자열을 상수 취급

### string

문자열을 위한 자료형  
문자열을 변수 취급

```
#include <iostream>
#include <string>
//string 클래스를 쓰기 위한 헤더 파일 include

int main() {
    std::string a;
    a = "a";
    std::string abcd = "abcd";

    return 0;
}
```

string은 문자열을 변수 취급하기 때문에 멤버 함수를 이용하여 **부분 삽입, 제거나 비교 등이 용이**해진다!

\*멤버 함수: 클래스 내에 정의된 함수

# <iostream> 라이브러리

02

cin >>

C++에서 입력을 위한 명령어

>> 라는 추출 연산자를 이용하여 값을 입력해줌

공백 문자를 받을 때까지 입력 가능

공백을 포함한 값을 입력 받고 싶다면?

cin.getline() 혹은 getline() 사용

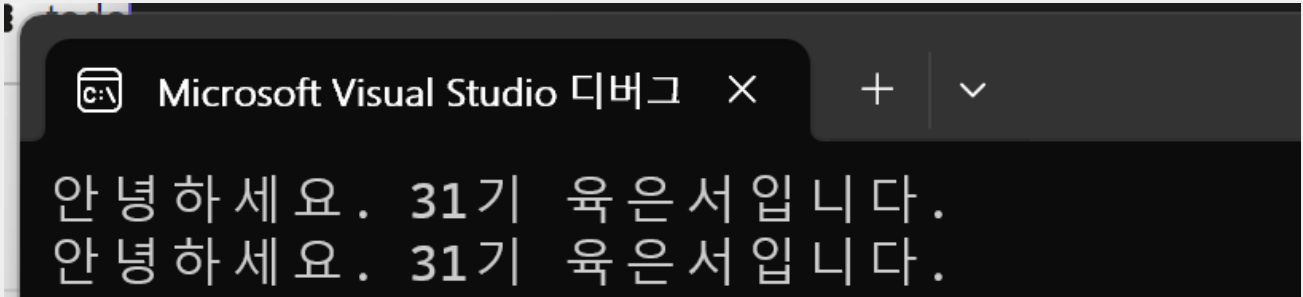
## char형의 공백 포함

\*이 방법은 두 번째 매개변수로 전달하는 범위 내에서만 입력 가능

```
main() {  
    //char형에서 한 줄 입력 받기  
    char swing[100];  
    std::cin.getline(swing, 100, '\n');
```

## string형의 공백 포함

```
main() {  
    //string형에서 한 줄 입력 받기  
    std::string swing;  
    std::getline(std::cin, swing); //이때 getline은 string 클래스
```



두 방법 모두  
공백을 포함한 문자열 입력하면(위)  
공백을 포함해서 저장 후 출력한다(아래)

# 다음 코드에서 안 배운 부분은?

---

20

```
#include <iostream>
#include <string>

int main() {
    std::string swing;
    std::getline(std::cin, swing);

    std::cout << swing;

    return 0;
}
```

# 다음 코드에서 안 배운 부분은?

---

21

std::

std::

std::

std::

# using namespace std;

22

```
int main() {  
    int cout = 8; //cout라는 변수 선언  
    cout << (cout << 1); //cout << 1 계산 결과값을 출력(cout)해주고 싶은데  
  
    return 0;  
}
```

\*cout << 1은 비트 연산자를 이용한 계산  
8 << 1은 16



Microsoft Visual Studio 디버거



# using namespace std;

23

namespace

식별자를 구별하기 위한 또 다른 식별자..

중복되는 식별자들을 구분하기 위한 내부 식별자

std

namespace중 하나, standard(표준)을 뜻함

```
#include <iostream>
#include <string>

int main() {
    int cout = 8; //cout라는 변수 선언
    std::cout << (cout << 1); //출력을 의미하는 cout를 std를 이용하여 식별

    return 0;
}
```

Microsoft Visual Studio 디버그

16

# using namespace std;

---

24

매번 std 적기 귀찮다! 코드에 중복되는 식별자가 없는데도 써야 할까?



# using namespace std;

25

매번 std 적기 귀찮다! 코드에 중복되는 식별자가 없는데도 써야 할까?

using 지시자

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

int main() {
    string swing;
    getline(cin, swing);

    cout << swing;

    return 0;
}
```

using namespace std;  
작성해주기

한 줄 주석

```
//한 줄 주석
```

// 사용

여러 줄 주석

```
/* 여러 줄에  
   걸쳐서  
   주석을  
   작성해야 한다면  
   이렇게  
   해보세요  
*/  
  
/* 당연히 한 줄도 가능 */
```

/\* \*/ 사용

01

+ -  
\* / %

산술 연산자

02

= += -=  
\*= /= %=

대입 연산자

03

++X X++  
--X X--

증감 연산자

이외에도 다양..

04

== !=  
> < >= <=

비교 연산자

05

&& || !

논리 연산자

06

& | ^  
~ << >>

비트 연산자

연산자	의미
+	덧셈
-	뺄셈
*	곱셈
/	나눗셈
%	나머지

연산자	의미
$x+=y$	$x=x+y$
$x-=y$	$x=x-y$
$x*=y$	$x=x*y$
$x/=y$	$x=x/y$
$x\%=y$	$x=x\%y$

연산자	의미
$y = x++$	$y = x$ 후 $x$ 값 1 증가
$y = ++x$	$x$ 값 1 증가 후 $y = x$
$y = x--$	$y = x$ 후 $x$ 값 1 감소
$y = --x$	$x$ 값 1 감소 후 $y = x$

연산자	의미
$x == y$	x와 y 값이 같은가
$x != y$	x와 y 값이 다른가
$x > y$	x가 y보다 큰가
$x >= y$	x가 y보다 크거나 같은가
$x < y$	x가 y보다 작은가
$x <= y$	x가 y보다 작거나 같은가

연산자	의미
&&	AND (모두 참)
	OR (하나만 참)
!	NOT (반전)



연산자	의미
&	비트 AND 연산
	비트 OR 연산
^	비트 XOR 연산
~	비트 NOT 연산
<<	왼쪽 시프트 연산
>>	오른쪽 시프트 연산

01

다음의 조건에서 input 방법 문서화

02

문서화 과제

01

다음의 조건에서 input 방법 문서화

오른쪽 프로그램 구현하기

프로그램 조건

- 모든 변수 자료형은 char
- 문자열을 a에 입력 받고 출력하는 코드 작성
- 띄어 쓴 두 문자열 a,b에 입력 받고 출력하는 코드 작성
- 공백이 포함된 문자열 c에 입력 받고 출력하는 코드 작성

코드를 작성하지만 티스토리 코드블럭 기능을 이용하여  
문서화 제출(깃허브 제출 X)

프로그래밍하면서 추가적으로 공부한 내용 정리하기

1. 문자열 입력 받기

a 입력: 홍길동

a의 내용은.. 홍길동

2. 띄어 쓴 두 문자열을 두 변수에 입력 받기  
단, 각각의 문자열 내에 공백 존재 X

a,b 입력: welcome swing

a의 내용은.. welcome

b의 내용은.. swing

3. 공백이 포함된 문자열 입력 받기

c 입력: 스윙 32기 홍길동

c의 내용은.. 스윙 32기 홍길동

hint.. 공백이 포함된 문자열 c 입력에서 자꾸 오류가 난다면 '입력 버퍼 비우기'에 대해 공부해보기..

02

## 문서화 과제

들어가야 하는 내용

- 1주차 스터디 자료 정리(6페이지까지의 이론 내용은 제외 가능)
- cstring 클래스와 string 클래스 비교
- cstring 클래스 주요 멤버 함수 정리 → 6개 이상 정리
- string 클래스 주요 멤버 함수 정리 → 10개 이상 정리

조건

- 모든 함수 정리는 오른쪽처럼 실습 화면 포함하기

```
Microsoft Visual Studio 디버그
str1 = helloworld
str2 = hi
str1.substr(2,1):l
str1.replace(2,1,str2):hehiloworld
str1.compare(str2):-1
str1.push_back('a'):hehiloworlda
str1.pop_back():hehiloworld

C:\Users\82104\source\repos\swing\x64\Debug\swing.exe(프로세스 21892개)이(가) 종료되었습니다(코드: 0)
이 창을 닫으려면 아무 키나 누르세요...
```

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

int main()
{
    string str1 = "helloworld";
    const char* str2 = "hi";

    //string 편집수정
    cout << "str1 = helloworld Wnstr2 = hi" << endl;
    cout << "str1.substr(2,1):" << str1.substr(2, 1) << endl;
    str1.replace(2, 1, str2);
    cout << "str1.replace(2,1,str2):" << str1 << endl;
    cout << "str1.compare(str2):" << str1.compare(str2) << endl;
    str1.push_back('a');
    cout << "str1.push_back('a'): " << str1 << endl;
    str1.pop_back();
    cout << "str1.pop_back():" << str1 << endl;

    return 0;
```

# THANK YOU

31기 육은서, 이시온, 황선영  
질문은 카톡으로 보내주세요~