Basic Computer Programming

Lecture 2

Electrical & Electronics Engineering
Chung-Ang University

Contents

- Understand the components that consist of the program.
- Understand the concept of comments.
- Understand the screen output.
- Make a program that receives inputs from a user.

The program we are going to make in this chapter

The explanation of the first program

```
#include <stdio.h>
int main(void)
{
        printf("Hello World!");
        return 0;
}
```

Including the header file.

Include the stdio.h file here

```
#include <stdio.h>
int main(void)
{
        printf("Hello World!");
        return 0;
}
```

Line change and indentation

It is classified by meaning by placing an empty line

```
#include <stdio.h>

int main(void)
{
        printf("Hello World!");
        return 0;
}
```

If the process is the same, indentations are made with tabs or spaces

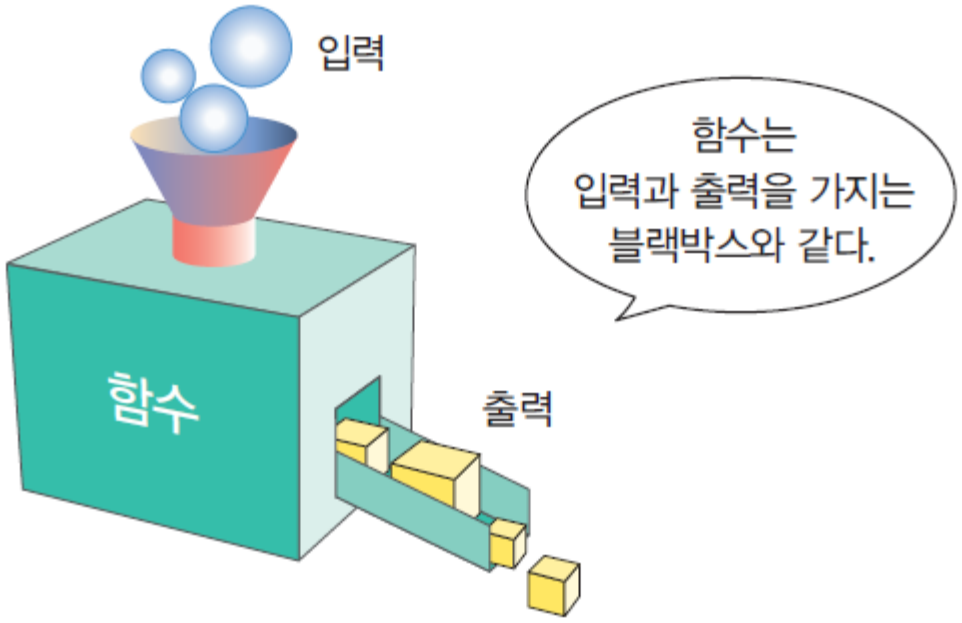
Function

- Sentences that perform a task must be included in the function.

```
#include <stdio.h>

int main(void)
{
        printf("Hello World!");
        return 0;
}
```

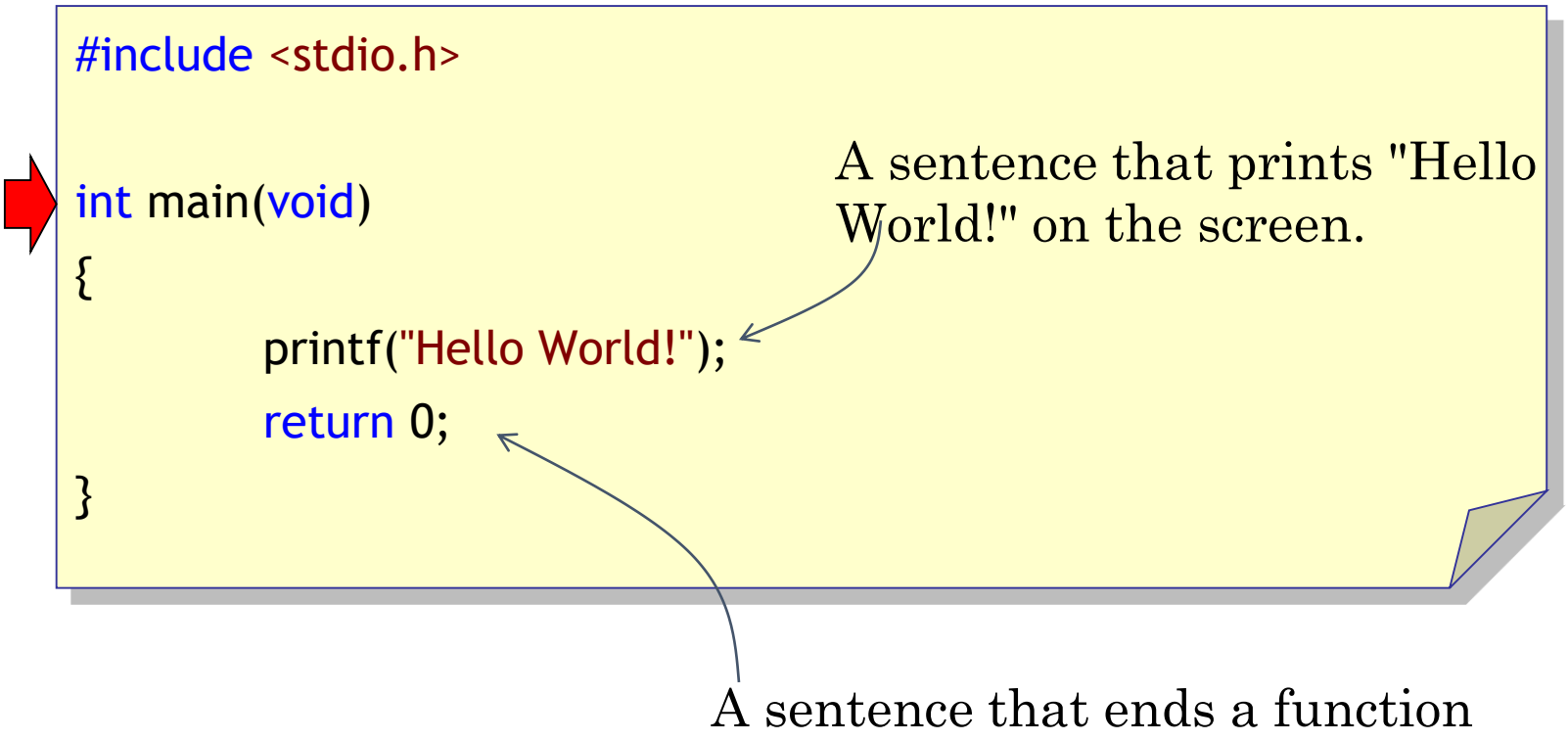
Function

- A function is a set of codes that perform a specific task.
- The function is like a black box that produces an output given input.

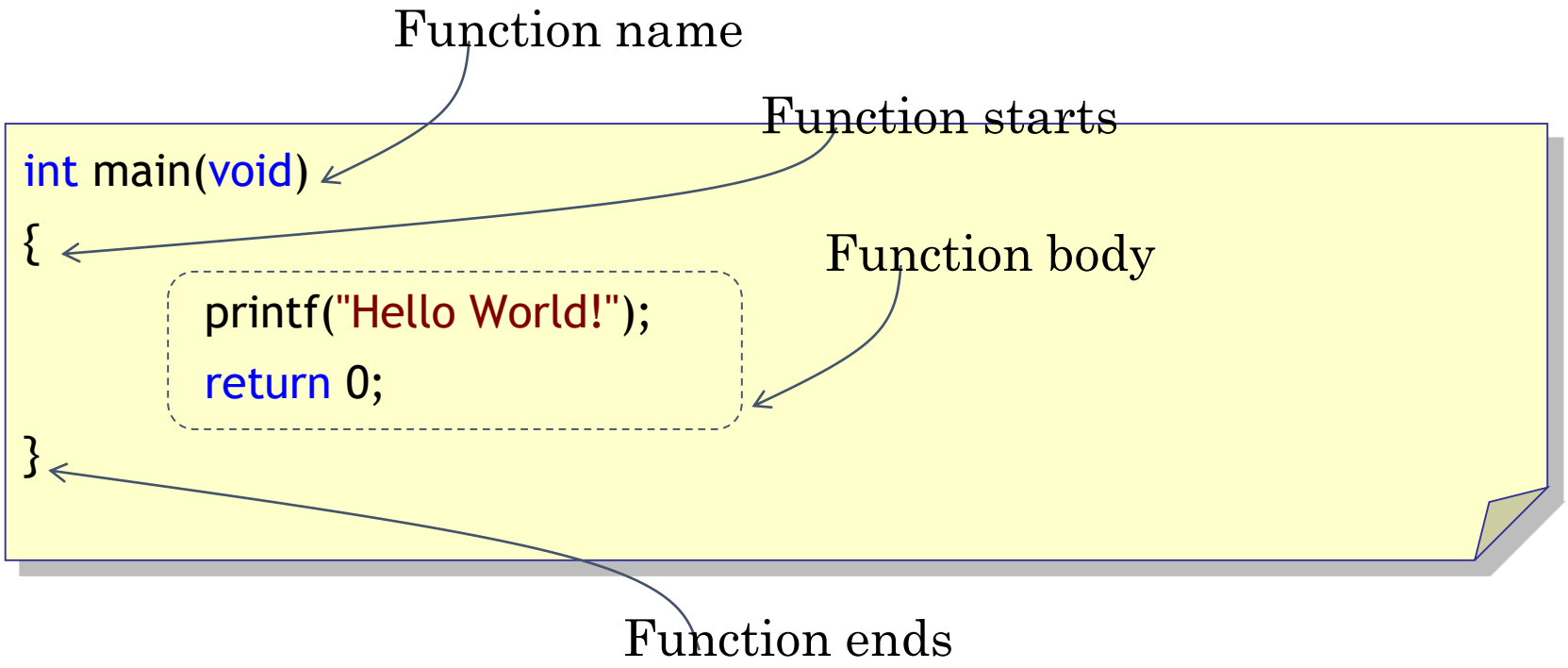


main() function

- The first function to run in program C.



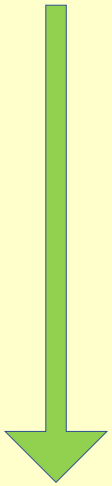
The components of the function



Commands

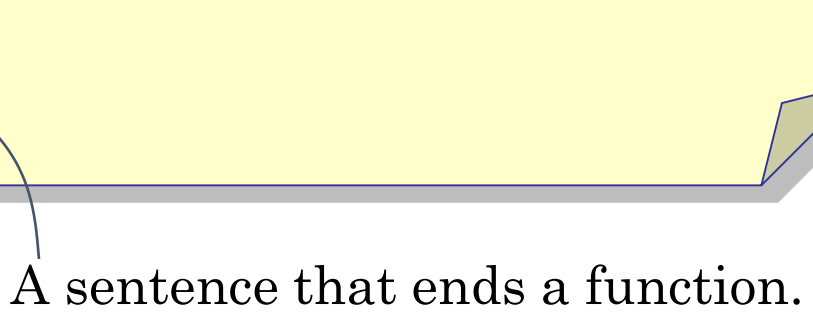
- The function consists of several commands
- commands are executed sequentially
- The command should end with ;

```
#include <stdio.h>

int main(void)
{
        printf("Hello World!");
        return 0;
}
```



Return sentence

```
#include <stdio.h>

int main(void)
{
        printf("Hello World!");
        return 0;
}
```



Comments

- Comment: Description of the program

```
/* 첫번째 프로그램 */
#include <stdio.h>

int main(void)
{
        printf("Hello World!");
        return 0;
}
```

Two ways to write comments

```
/* 작성자: 홍길동
작성날짜: 2017년 3월 1일
작성목적: 컴파일러 테스트
*/
```

```
// 이 줄은 전체가 주석이다.
printf("Hello World!"); // 문자열을 화면으로 출력
```

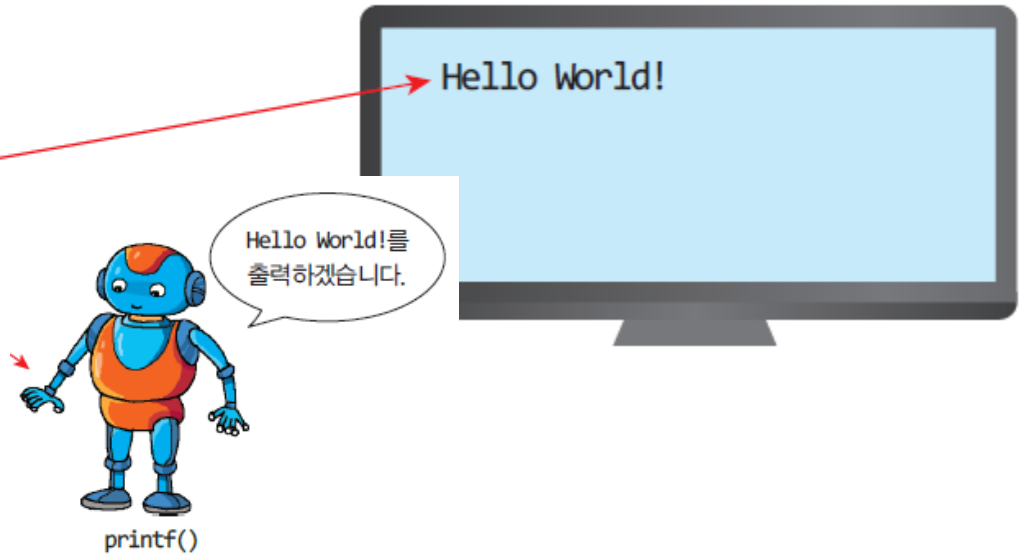
Interim check

1. What kind of performance does the #include do?
2. Which symbol should be at the end of every sentence?
3. What role does a comment play?

Print function printf()

- printf() is a library function that outputs a string in double quotes on the screen.

```
int main(void)
{
   printf("Hello World!");
   return 0;
}
```



Application #1

- Let's make a program with the following output.

The first version

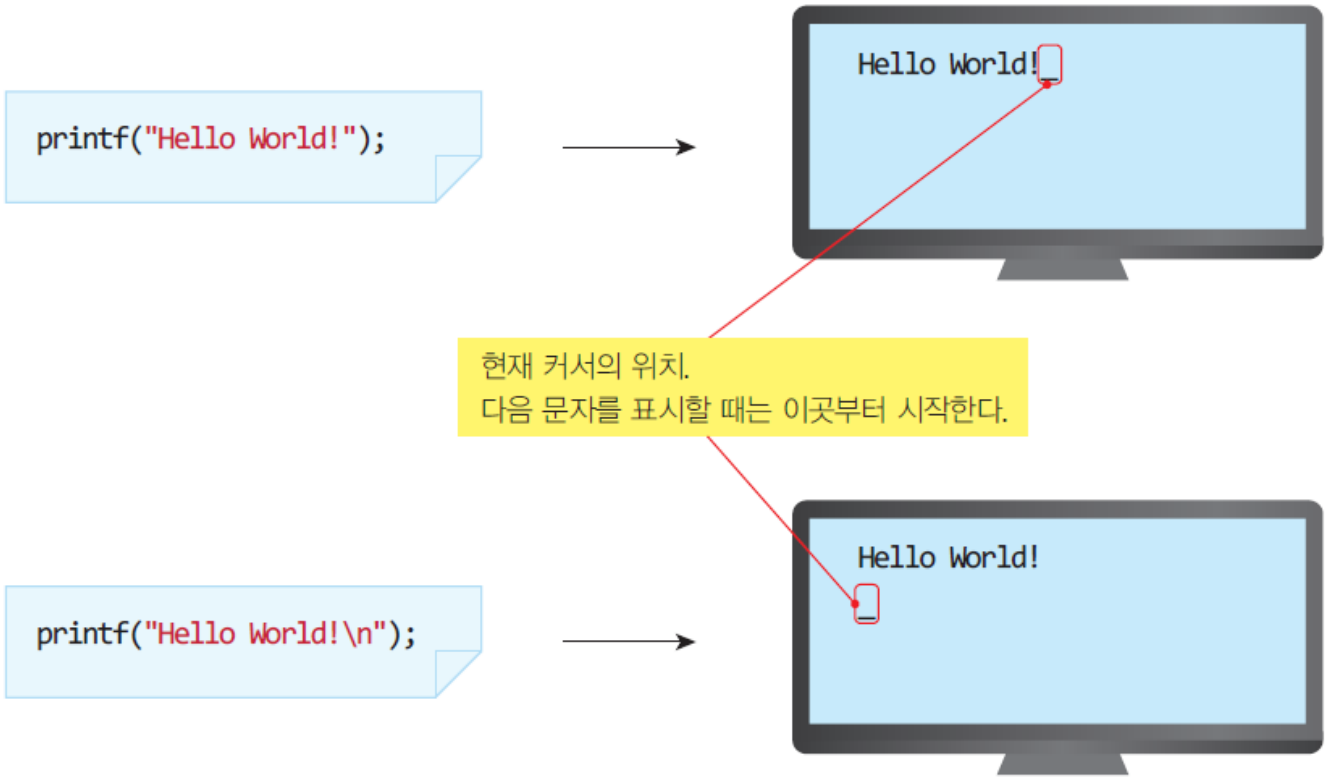
- Use the fact that sentences are executed sequentially

```
/* 첫 번째 프로그램의 응용 */
#include <stdio.h>
int main(void)
{
        printf("Hello World!");
        printf("Kim ChulSoo");
        return 0;
}
```

It's not the result we want!

The line-switching character \n

- The line-switching character \n moves the cursor to the next line on the screen



The changed program

- Including the line-changing characters results in what we wanted

```
/* 첫 번째 프로그램의 응용 */
#include <stdio.h>
int main(void)
{
        printf("Hello World!\n");
        printf("Kim ChulSoo\n");
        return 0;
}
```

```
Microsoft Visual Studio 디버그 콘솔
Hello World!
from ChulSoo
```

A program printing multiples of three

- A part of multiples of three are printed

A program printing multiples of three

- A part of multiples of three are printed

```c
/* 첫 번째 프로그램의 응용 */
#include <stdio.h>

int main(void)
{
        printf("3 X 1 = 3\n");
        printf("3 X 2 = 6\n");
        printf("3 X 3 = 9\n");
        return 0;
}
```

Interim check

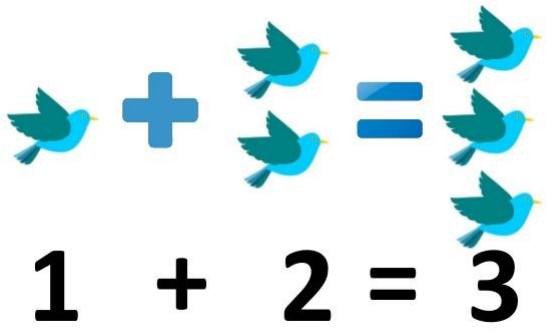
1. What is the role of "\n" in the character of line change?
2. In what order are the sentences in the main() function executed?
3. Let's write a program that prints multiple of 9.

The form of typical programs

- The data is received (input step), processed (processing step), and the result is output (output step) on the screen.

Our goal

- Let's write a program that calculates the sum of two integers



Source code

```c
/* 두 개의 숫자의 합을 계산하는 프로그램 */
#include <stdio.h>

int main(void)
{
        int x;                  // 첫 번째 정수를 저장할 변수
        int y;                  // 두 번째 정수를 저장할 변수
        int sum;                // 두 정수의 합을 저장하는 변수

        x = 100;
        y = 200;

        sum = x + y;
        printf("두수의 합 = %d \n", sum);

        return 0;
}
```

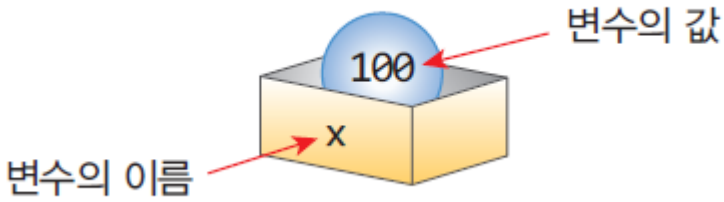
Variable

- Variable: Memory space used for the purpose of temporarily storing data used by a program

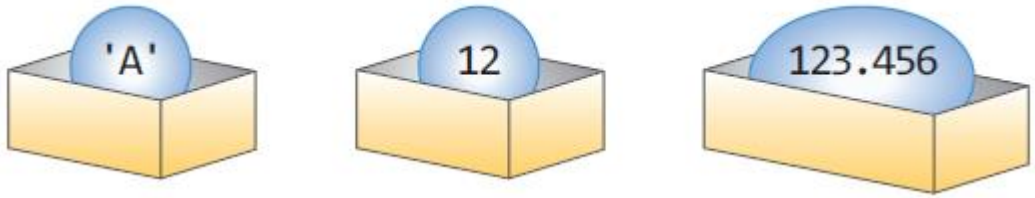
```
int x;     // 첫번째 정수를 저장하는 변수
int y;     // 두번째 정수를 저장하는 변수
int sum; // 두 정수의 합을 저장하는 변수
```

Types of variables

- Variables can be thought of as boxes containing data.

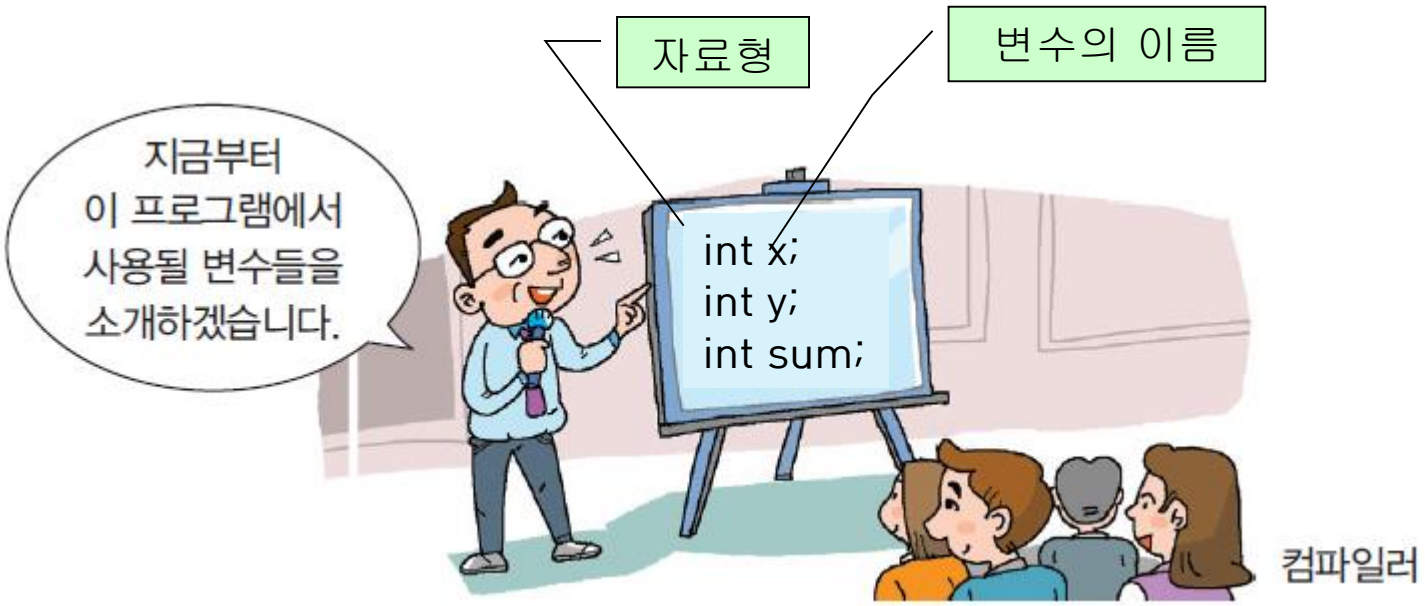


- There are several types of variables depending on the type of data.



Declaration of variables

- Variable Declaration: To inform the compiler in advance of what type of variable is used.



Declaration of variables

Syntax 2.1 변수 선언

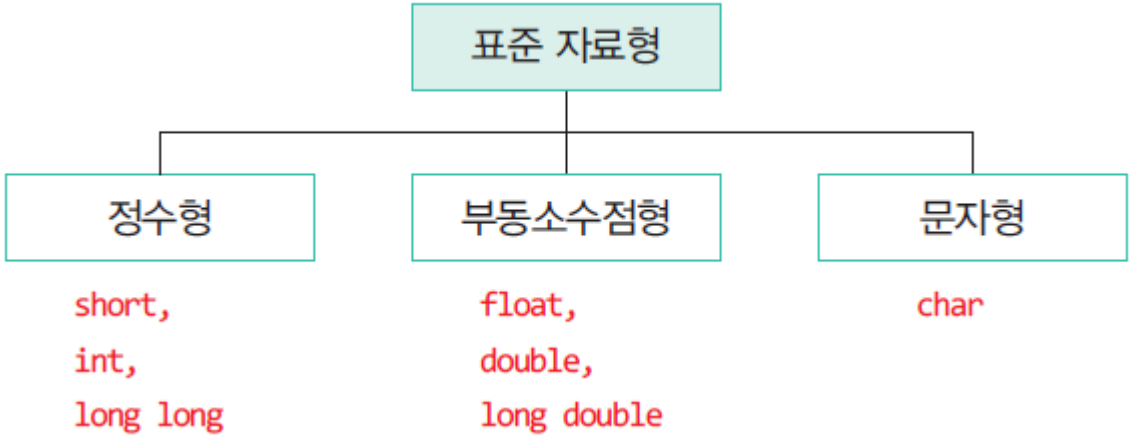
형식 자료형 변수이름;

예

```
int i;
float sum;
double x, y;
```

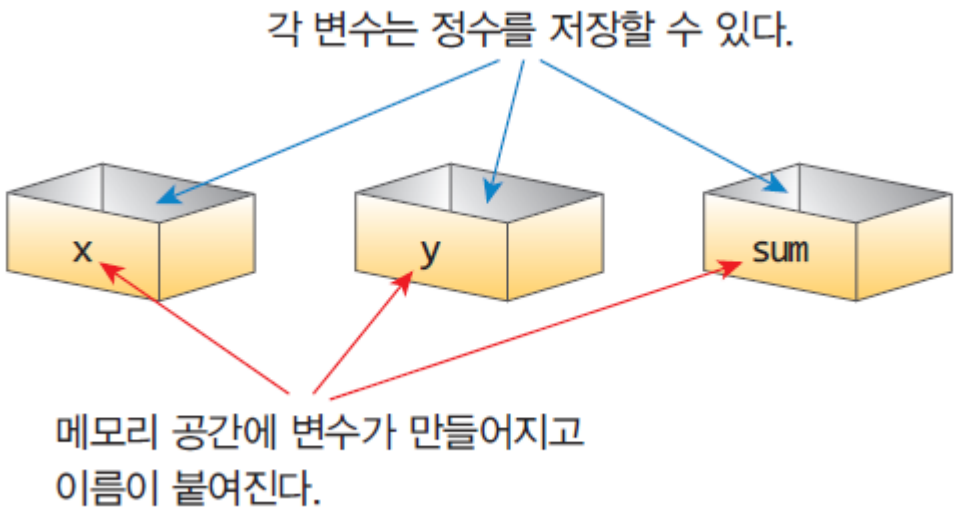
Data type

- Data type: To specify whether the data to store the variable is an integer or a real number, or what else



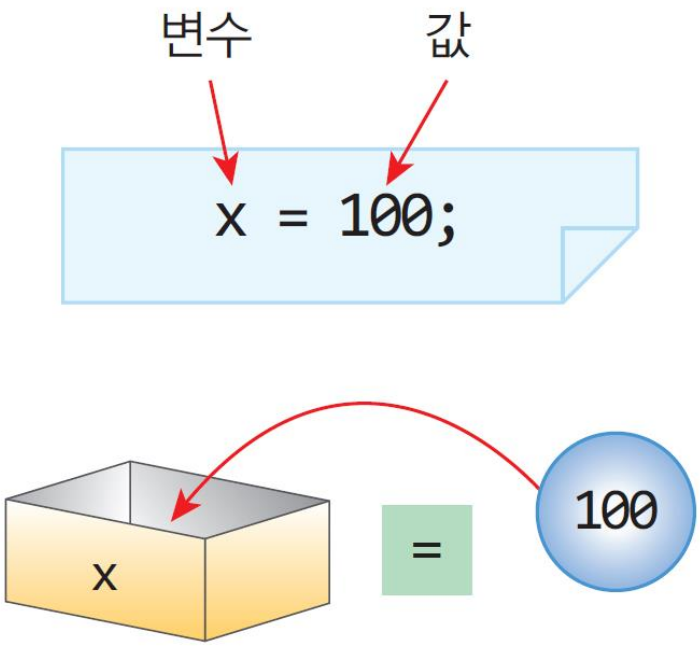
Declaration of variables

```
int x;     // 첫번째 정수를 저장하는 변수
int y;     // 두번째 정수를 저장하는 변수
int sum; // 두 정수의 합을 저장하는 변수
```



Assignment operation

- An operation to store values in variables.
- It is also called an allocation operation.

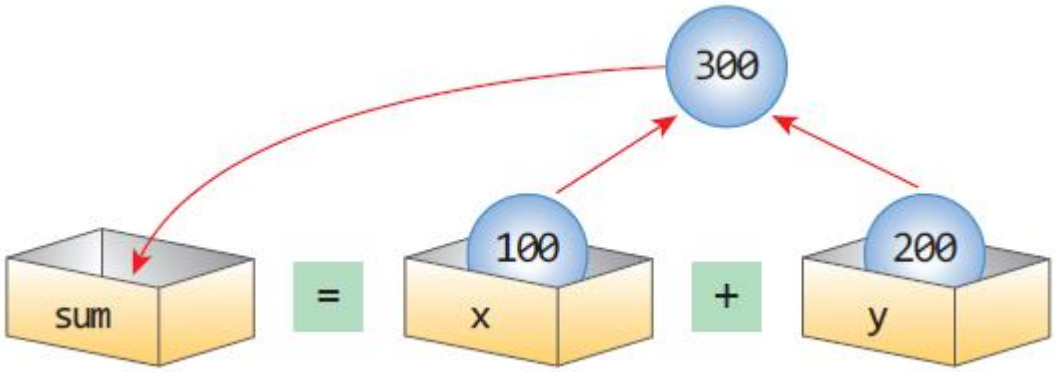


Arithmetic operation

연산	연산자	C 수식
덧셈	+	x + y
뺄셈	-	x - y
곱셈	*	x * y
나눗셈	/	x / y
나머지	%	x % y

Arithmetic operation

```
sum = x + y;
```



printf()

Syntax 2.2 printf() 사용하기

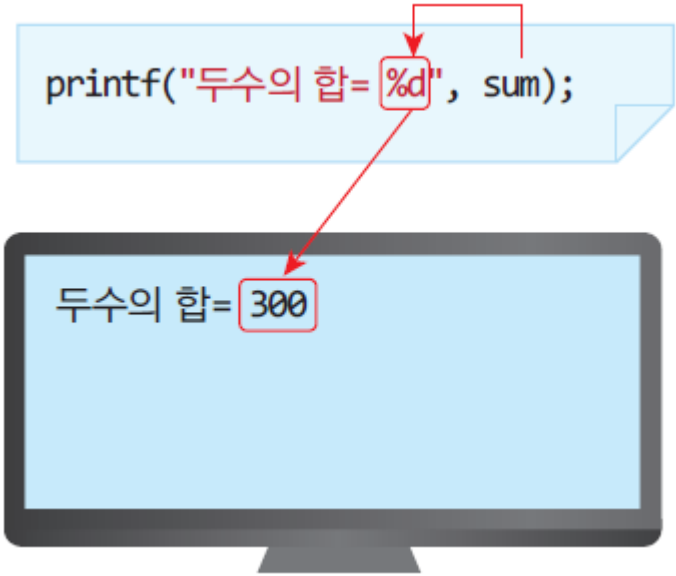
형식 printf(형식지정자, 변수);

예 printf("%d", sum);

The form of printf()

형식 지정자	의미	형태
%d	정수 형태로 출력	100
%f, %lf	실수 형태로 출력	3.141592
%c	문자 형태로 출력	A
%s	문자열 형태로 출력	Hello

The printing process of printf()



Review

```c
/* 두 개의 숫자의 합을 계산하는 프로그램 */
#include <stdio.h>

int main(void)
{
        int x;                  // 첫 번째 정수를 저장할 변수
        int y;                  // 두 번째 정수를 저장할 변수
        int sum;                // 두 정수의 합을 저장하는 변수

        x = 100;
        y = 200;

        sum = x + y;
        printf("두수의 합 = %d \n", sum);

        return 0;
}
```

Addition program #2

- Let's get input from a user

```c
// 사용자로부터 입력받은 정수 2개의 합을 계산하여 출력
// 비주얼 스튜디오 사용자라면 다음 문장이 필요하다.
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main(void)
{
	int x; // 첫 번째 정수를 저장할 변수
	int y; // 두 번째 정수를 저장할 변수
	int sum; // 2개의 정수의 합을 저장할 변수

	printf("첫 번째 숫자를 입력하시오:"); // 입력 안내 메시지 출력
	scanf("%d", &x); // 하나의 정수를 받아서 x에 저장

	printf("두 번째 숫자를 입력하시오:"); // 입력 안내 메시지 출력
	scanf("%d", &y); // 하나의 정수를 받아서 x에 저장

	sum = x + y; // 변수 2개를 더한다.
	printf("두수의 합= %d \n", sum); // sum의 값을 10진수 형태로 출력

	return 0; // 0을 외부로 반환
}
```

scanf()

Syntax 2.3 scanf()

형식 scanf(형식지정자, &변수);

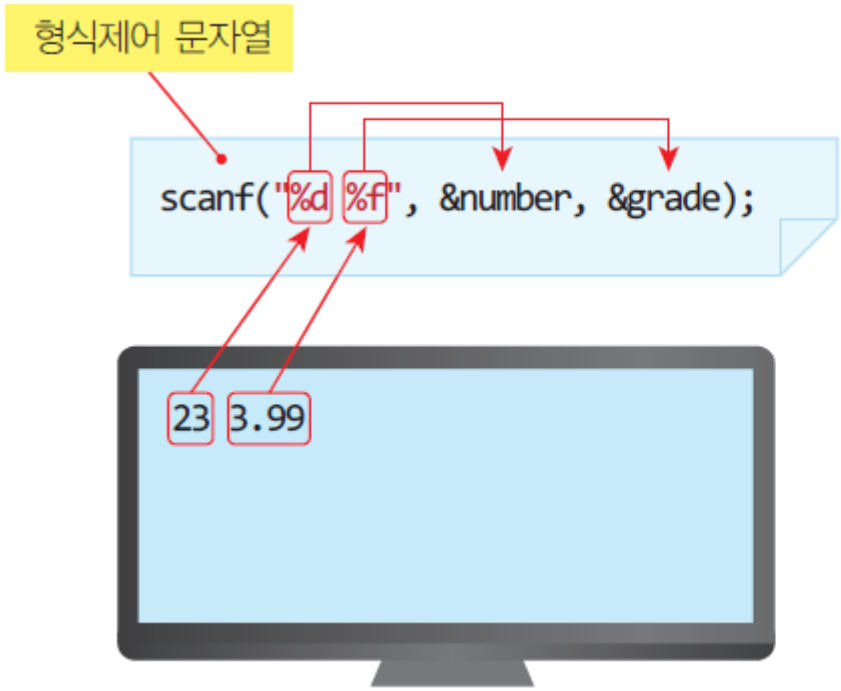
예 scanf("%d", &x);

Formatting designator used in scanf()

- Similar to printf()

형식 지정자	의미	형태
%d	정수	100
%f	실수(float)	3.14
%lf	실수(double)	3.141592
%c	문자	A
%s	문자열	Hello World!

scanf()


형식제어 문자열
scanf("%d %f", &number, &grade);
23 3.99


형식 지정자의
개수와 변수의 개수와 순서는
같아야 합니다.

The meaning of &

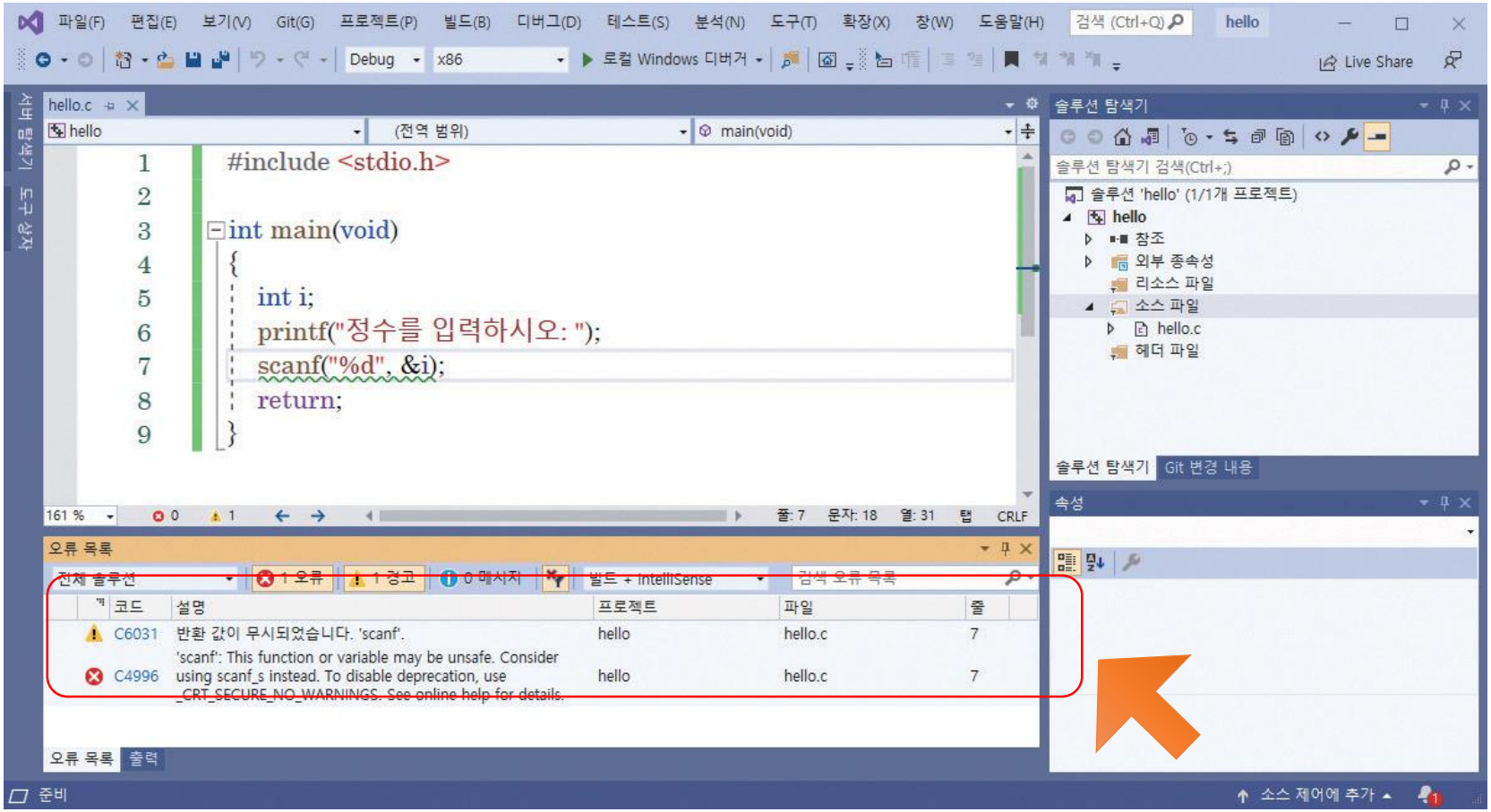
- & Operator: Operator that calculates the address of a variable.
- The address of the variable is required to store the value in the variable.

Input numbers

```
float ratio = 0.0;
scanf("%f", &ratio);

double scale = 0.0;
scanf("%lf", &scale);
```

Notice!

scanf() function in Visual Studio

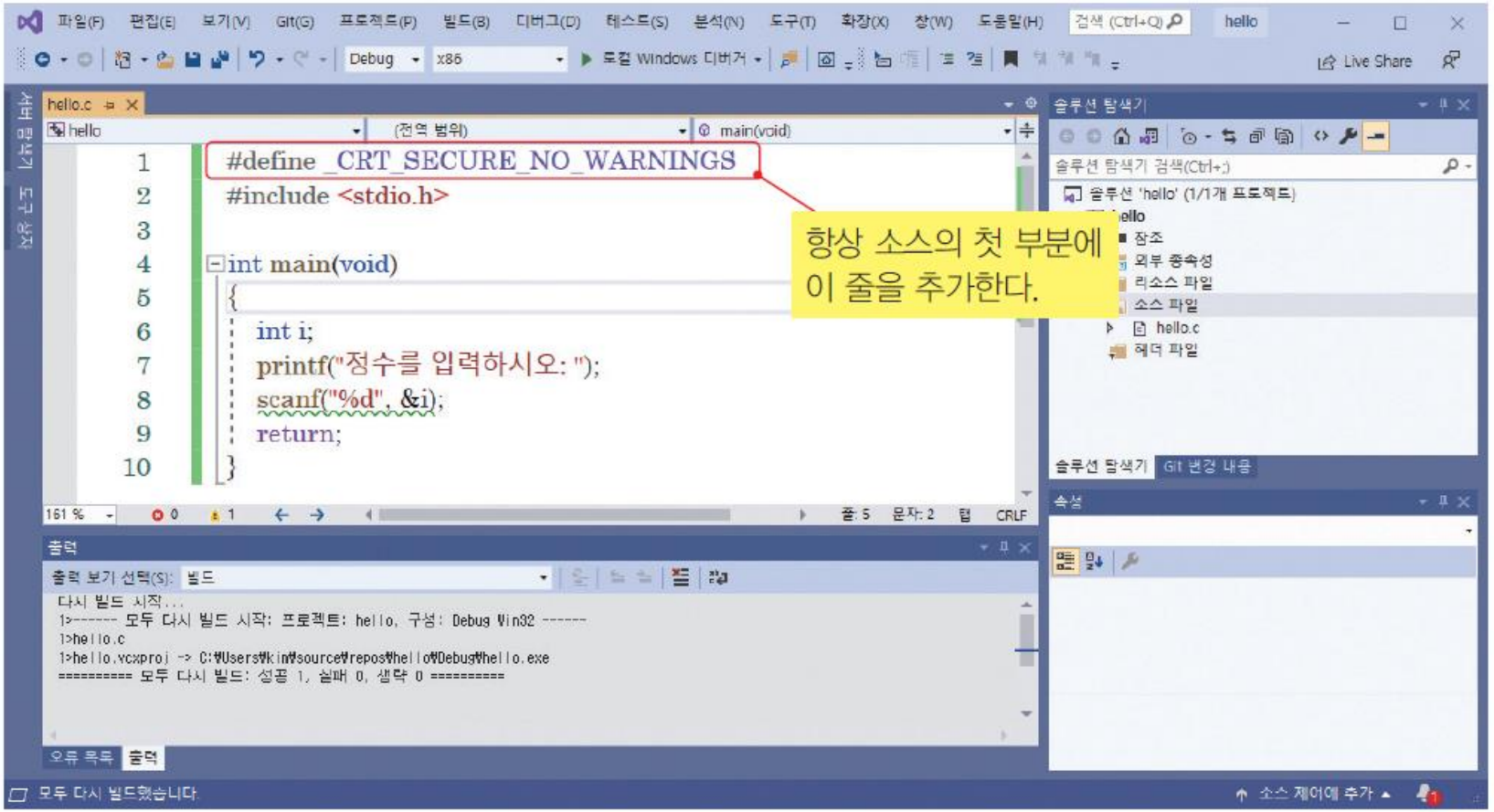


Scanf() function error in Visual Studio 2019

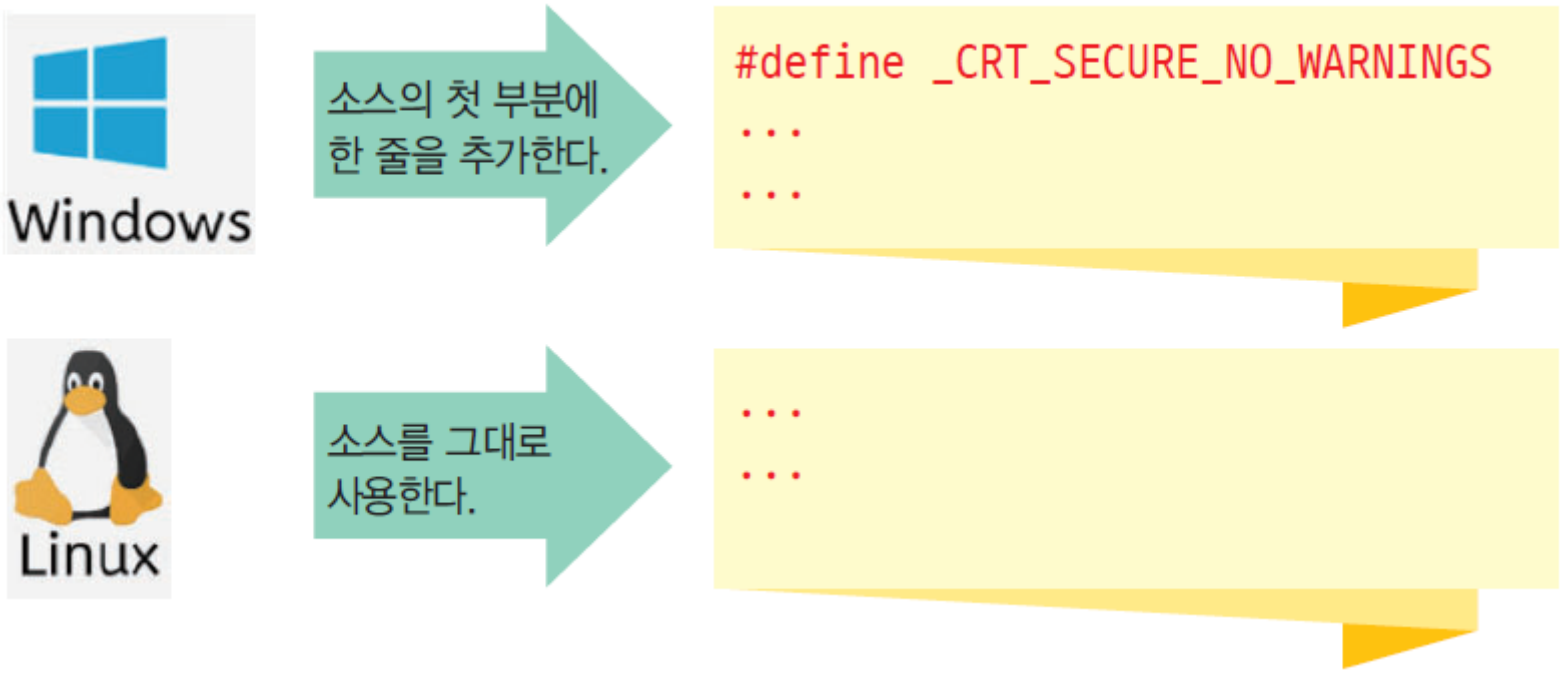
- An error that notice to use scanf_s() instead because scanf() is not safe.
- Safe functions that attach _s to existing functions, such as scanf_s(), are optional standards of C11 published in 2011 (Annex K).
- However, because it is an optional standard, it is still not actively introduced in other compilers, including gcc, except for visual studios (the field application report evaluated it as "not very beneficial." It was recommended to be deleted from the following standards)

- **Conclusion: Define _CRT_SECURE_NO_WARNINGS at the very beginning of the source code and use the existing functions as they are.**

Scanf() function error in Visual Studio 2019



Scanf() function error in Visual Studio 2019

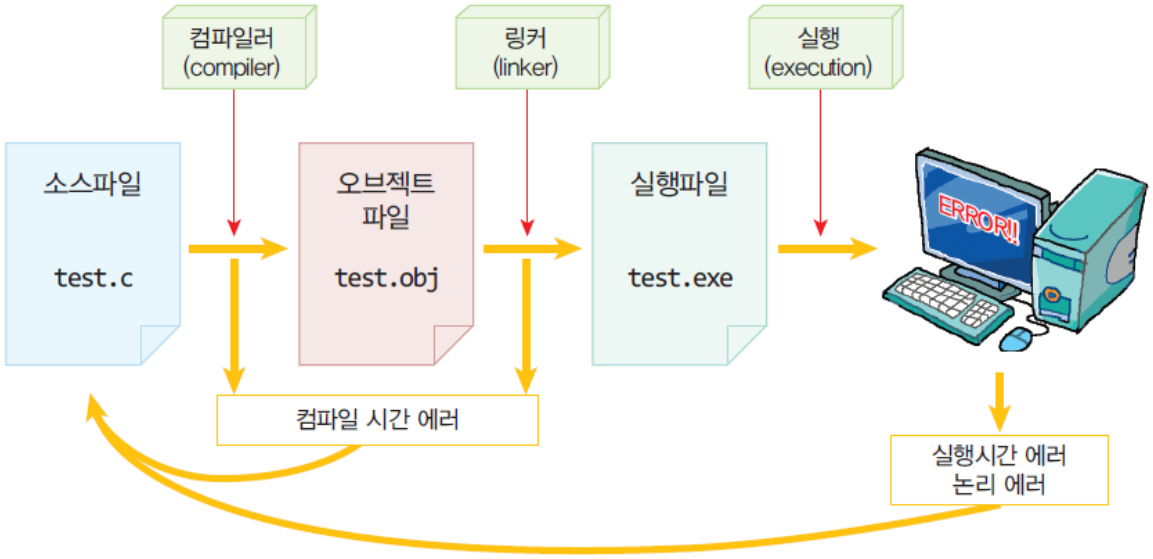


Types of errors.

- Error: Serious error that cannot be compiled or linked
- Warning: Compilation, links are possible and executable, but minor errors that can cause potential problems.

Other errors

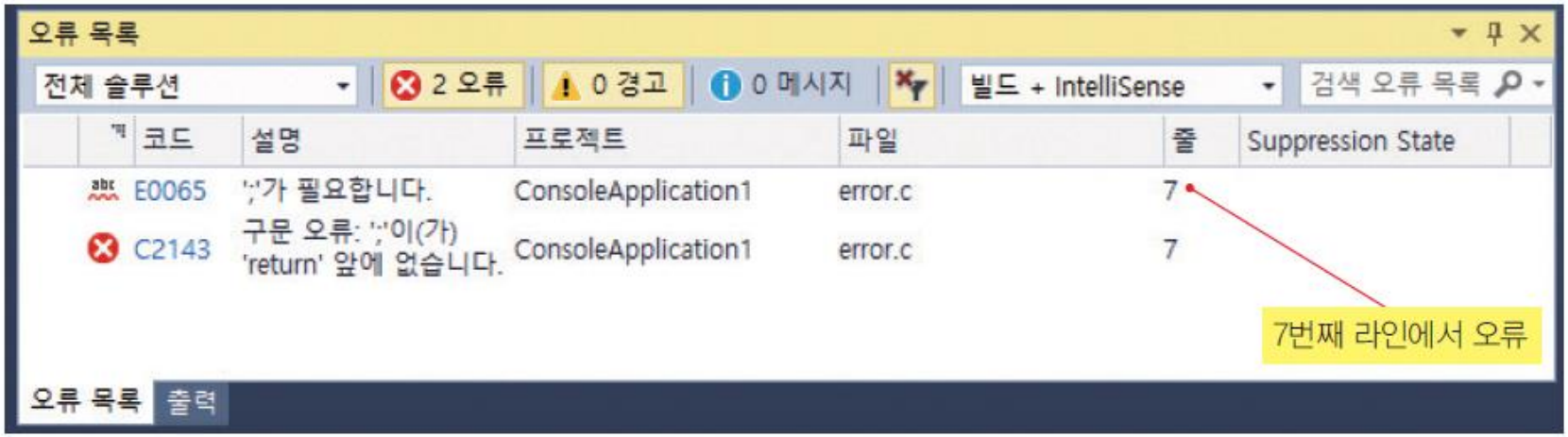
- Compilation time error: mostly grammatical error
- Execution time error: Error occurring during execution
- Logic error: something that goes wrong logically and doesn't come out as intended.



An example of errors

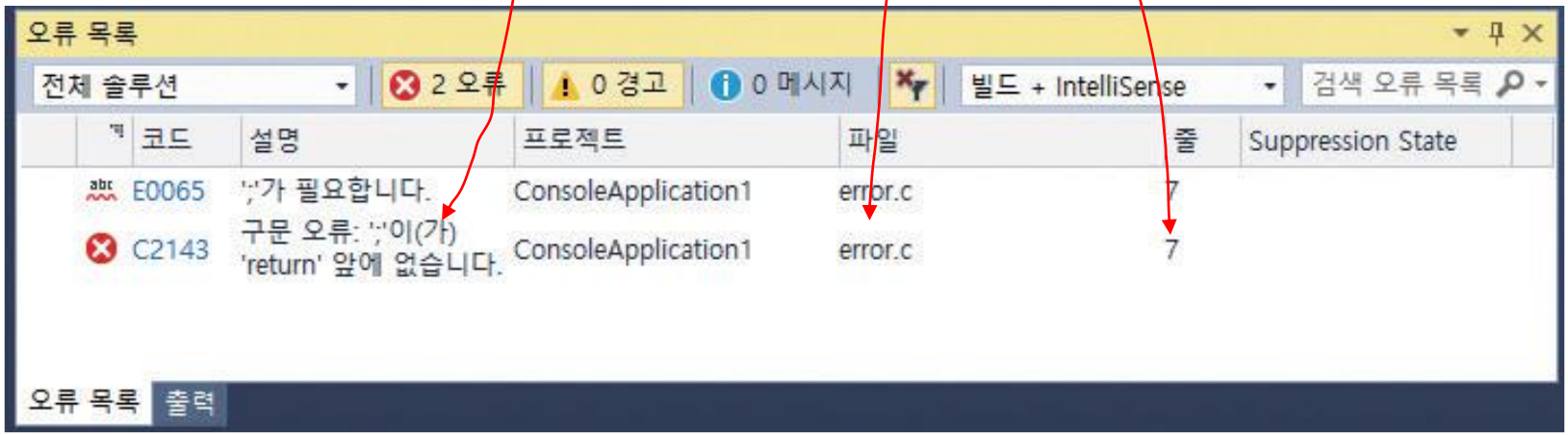
```c
// 에러가 발생하는 프로그램
#include <stdio.h>

int main(void)
{
        printf("Hello World!\n")
        return 0;
}
```

The semicolon is missing at the end of the sentence.



What if there's an error?

- Let's take a closer look at the next one.
  - Name of the file where the error was found.
  - The number of the line where the error was found.
  - Contents of the error found (Description)



Interim check

1. Let's classify the errors according to the severity.
2. What error does the written program fall under if it does not obey the grammar of the C language?

Lab: Math Operations

- Let's write a program that receives two integers from the user and outputs them on the screen by calculating +, -, *, /.

Lab: Math Operations

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main(void)
{
	int x;// 첫 번째 정수를 저장할 변수
	int y;// 두 번째 정수를 저장할 변수
	int result;// 연산의 결과를 저장할 변수

	printf("첫 번째 숫자를 입력하시오:");// 입력 안내 메시지 출력
	scanf("%d", &x);// 하나의 정수를 받아서 x에 저장

	printf("두 번째 숫자를 입력하시오:");// 입력 안내 메시지 출력
	scanf("%d", &y);// 하나의 정수를 받아서 y에 저장
```

Lab: Math Operations

```
    result = x + y;// 덧셈
    printf("두수의 합= %d \n", result);
    result = x - y;// 뺄셈
    printf("두수의 차= %d \n", result);
    result = x * y;// 곱셈
    printf("두수의 곱= %d \n", result);
    result = x / y;// 나눗셈
    printf("두수의 몫= %d \n", result);

    return 0;
}
```

도전문제

상품 2개의 가격을 받아서 상품의 평균 가격을 계산하는 프로그램을 작성해보자. 평균 가격은 (p1+p2)/2.0으로 계산할 수 있다. 여기서 p1은 첫 번째 상품의 가격이고, p2는 두 번째 상품의 가격이다.

Lab: Calculation of travel expenses

- Total travel expenses = Flight ticket price + (hotel price + daily allowance) * Number of days of stay

```
Microsoft Visual Studio 디버그 콘솔
여행은 몇박인가요?: 3
항공권 가격: 1000000
호텔 1박 가격: 200000
하루에 필요한 용돈: 50000
================================
총 여행 비용: 1750000
================================
```

Lab: Calculation of travel expenses

```
#include <stdio.h>

int main(void)
{
        int sum;                // 총액을 저장하는 변수
        int price;      // 각 항목의 가격을 저장하는 변수
        int nights;     // 몇 박인지를 저장하는 변수

        sum = 0;                // 0으로 초기화한다.
        printf("여행은 몇박인가요?: ");
        scanf("%d", &nights);

        printf("항공권 가격: ");
        scanf("%d", &price);
        sum = sum + price;

        printf("호텔 1박 가격: ");
        scanf("%d", &price);
        sum = sum + days*price;
```

Lab: Calculation of travel expenses

```
        printf("하루에 필요한 용돈: ");
        scanf("%d", &price);
        sum = sum + nights*price;

        printf("================================\n");
        printf("총 여행 비용: %d \n", sum);
        printf("================================\n");

        return 0;
}
```

Q & A