

---

# **C** 프로그래밍 및 실습

## **1. C** 프로그래밍 시작하기

세종대학교

---

# 목차

---

- 1) 프로그래밍 개요
- 2) C 프로그램 작성하기
- 3) 첫 프로그램 분석
- 4) 출력 기초
- 5) 입력 기초

# 1) 프로그래밍 개요

---

- 프로그램과 프로그래밍

- 프로그램

- ✓ 운동회의 순서나 음악회의 연주 곡목 순서 등 미리 짜 놓은 것
    - ✓ 예) TV 프로그램

- 컴퓨터 프로그램 (program)

- ✓ 컴퓨터가 수행해야 하는 일의 순서와 방법을 나타냄
    - ✓ 순서에 따라 한 번에 하나씩 실행되는 명령어로 구성

- 프로그래밍

- ✓ 프로그래밍 언어를 이용하여 프로그램을 개발하는 행위

# 1) 프로그래밍 개요

---

- 프로그래밍 언어와 C 언어

- 프로그래밍 언어

- ✓ 기계어: 컴퓨터가 이해할 수 있는 언어로 0과 1의 이진 코드로 구성
- ✓ 어셈블리어: 기계어의 명령어를 사람이 이해할 수 있는 단어로 일대일로 대응시킨 언어
- ✓ 고급언어: 사람의 사고 체계에 적합하게 만들어진 언어  
C 언어, C++, C#, Java, Python 등

- C 언어

- ✓ 1970년대에 개발된 프로그래밍 언어로, Unix 운영체제 개발에 활용
- ✓ 현재까지도 많이 활용되는 언어

# 1) 프로그래밍 개요

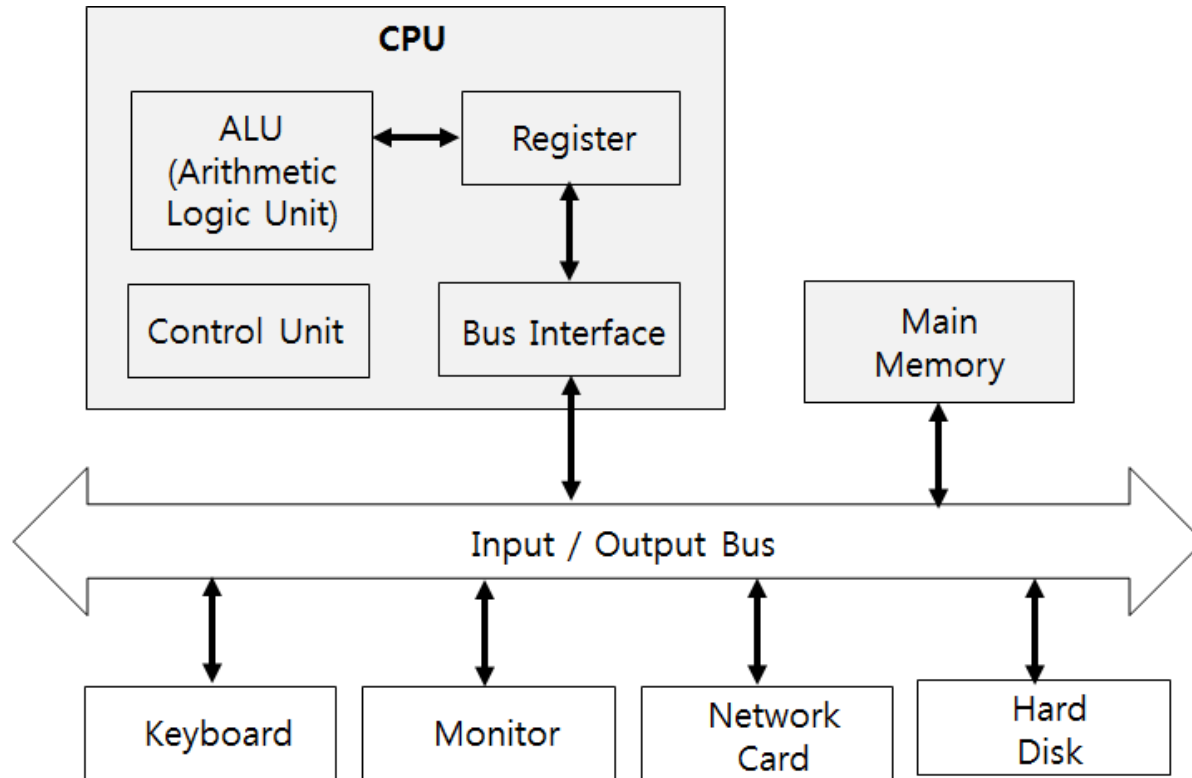
---

- **C 언어의 장단점**

- 이식성이 높다( C is a highly portable language).
- 구조화된 프로그래밍 언어이다.
- 하드웨어의 미세한 조정까지 가능하다.
- 고급 및 저급 수준의 특징을 동시에 포함하고 있다.
- 저급 수준의 특징을 가지다 보니 타 고급 언어에 비해서 프로그램 이해가 어렵다
- 자유도가 높다 (장점이자 단점)

# 1) 프로그래밍 개요

- 컴퓨터 시스템



# 1) 프로그래밍 개요

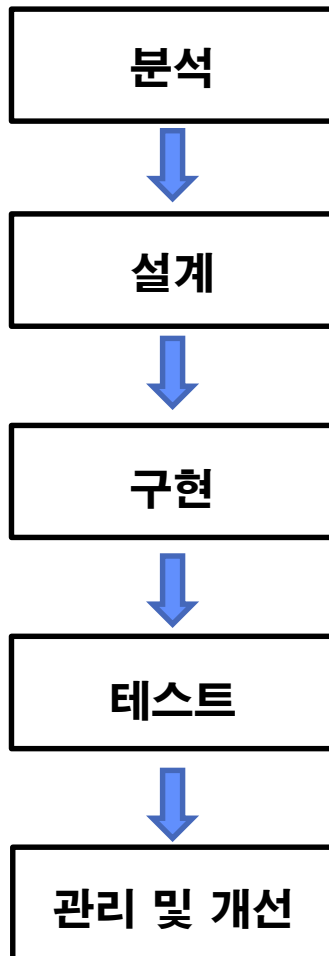
---

- 비트와 바이트

- 컴퓨터에서 처리되는 정보는 기본적으로 0과 1로 구성된 이진수로 표현
- 비트(bit): 한 자리 이진수로 표현되는 정보 단위
- 바이트(byte): 8개의 비트를 묶은 정보 단위 ( 1 바이트 = 8 비트)
  - ✓ 1 바이트는  $2^8 = 256$  개의 상태 표현 가능

# 1) 프로그래밍 개요

---



## ■ 소프트웨어 개발 과정

- 소프트웨어 시스템을 개발하는 설계자들은 개발 단계를 따름
- 요구사항을 분석한 후 프로그램을 설계
- 프로그램을 읽기 쉽고 바꾸기 쉽게 설계하지 않으면 프로그램을 개선하거나 추가하기가 어려움
- 요구사항을 분석하여 테스트 절차서 작성



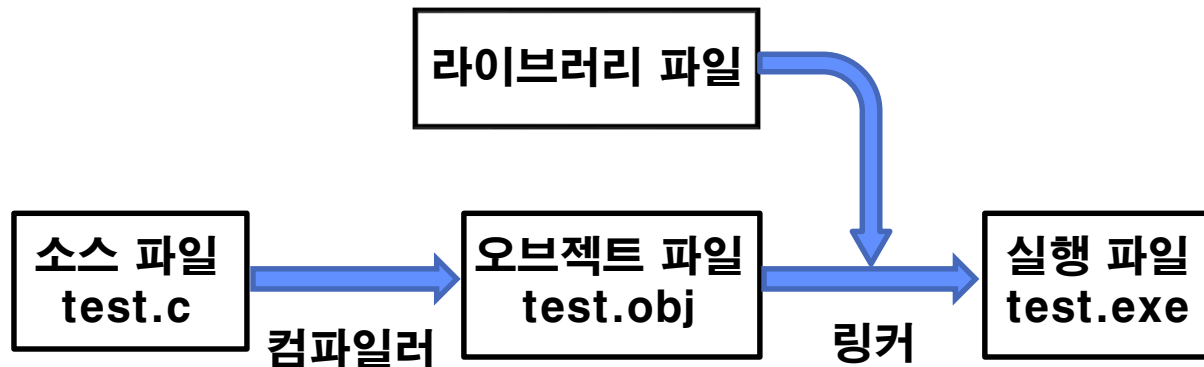
# 목차

---

- 1) 프로그래밍 개요
- 2) **C 프로그램 작성하기**
- 3) 첫 프로그램 분석
- 4) 출력 기초
- 5) 입력 기초

## 2) C 프로그램 작성하기

- C 프로그램이 만들어 지는 과정
  - 소스 파일 : 컴퓨터가 수행해야할 일을 명세한 텍스트 파일
  - 오브젝트 파일 : 소스 파일을 기계어로 번역된 것
  - 라이브러리 파일: 널리 사용되는 기능을 미리 구현해 놓은 파일
  - 실행 파일 : 오브젝트, 라이브러리, 시동 코드를 결합한 것
- 컴파일: 소스 파일을 기계어로 변환하는 것



## 2) C 프로그램 작성하기

---

- **Visual Studio (VS)**

- MS 사의 통합 개발 환경 (IDE, 프로그램 개발에 관련된 모든 작업을 처리하는 환경)
- C 언어 뿐만 아니라, C++, C# 등 다른 프로그래밍 언어도 지원

- **VS를 이용한 C 프로그램 개발 단계**

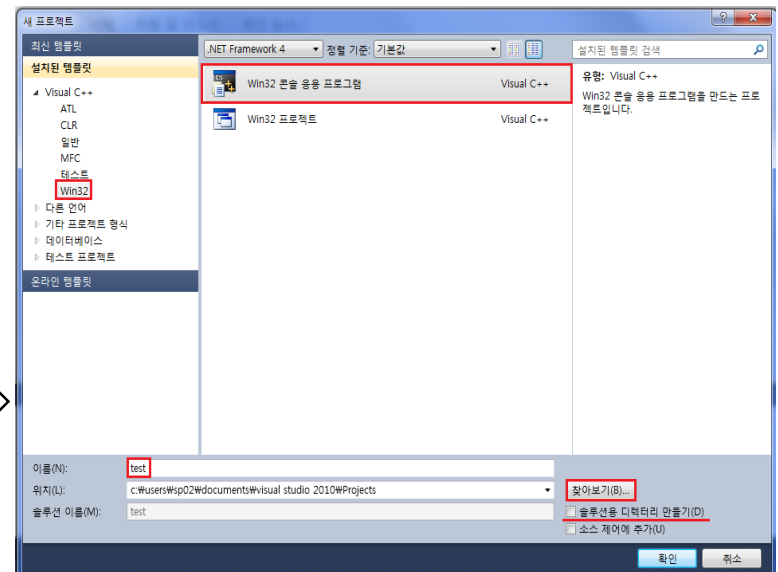
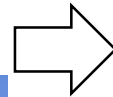
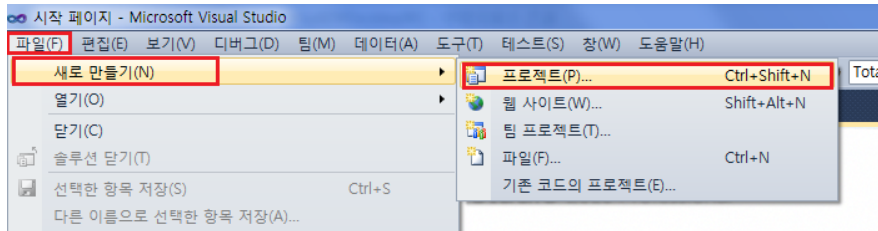
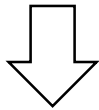
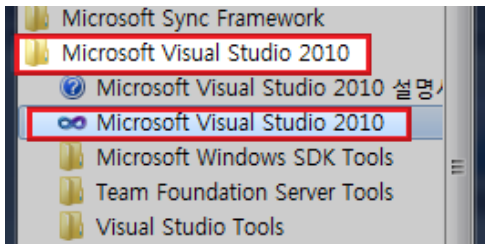
1. 프로젝트 생성
2. 소스 코드 작성
3. 프로그램 빌드(실행 파일 생성)
4. 프로그램 실행

(참고) 강의 자료에서는 VS 2010 버전으로 설명  
교재에는 VS 2017 버전에 대한 설명 수록

## 2) C 프로그램 작성하기

### ■ 1단계 - 프로젝트 생성 (1)

- Visual Studio 실행 후 새프로젝트 생성
  - ✓ 파일 → 새로만들기 → 프로젝트
  - ✓ (Win32 선택) → Win32 콘솔 응용 프로그램 선택
  - ✓ 프로젝트 이름 입력



## 2) C 프로그램 작성하기

### ▪ 1단계 - 프로젝트 생성 (2)

- 마법사 화면에서

- ✓ '다음>' 선택 → 콘솔 응용프로그램 선택, 빈 프로젝트 체크 → 마침

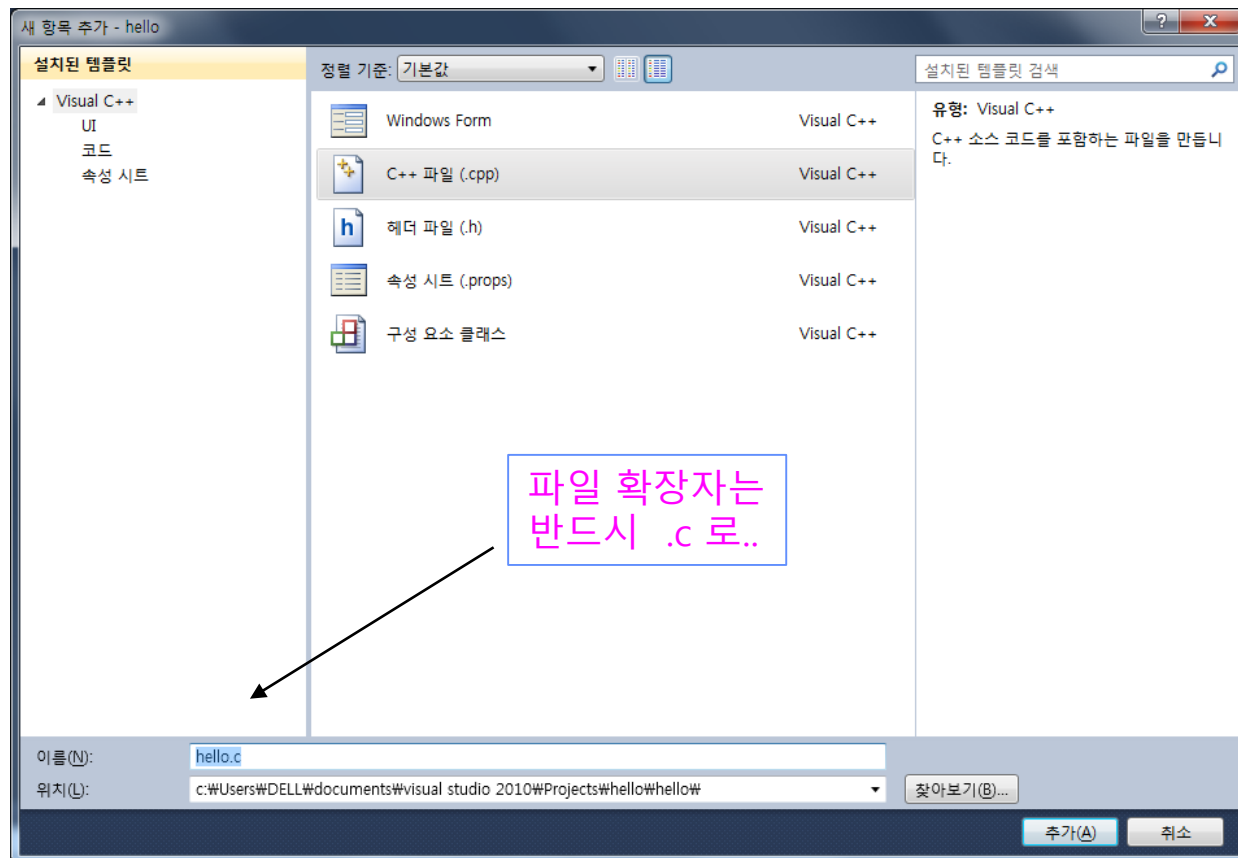


## 2) C 프로그램 작성하기

### ■ 2단계 - 소스코드 작성 (1)

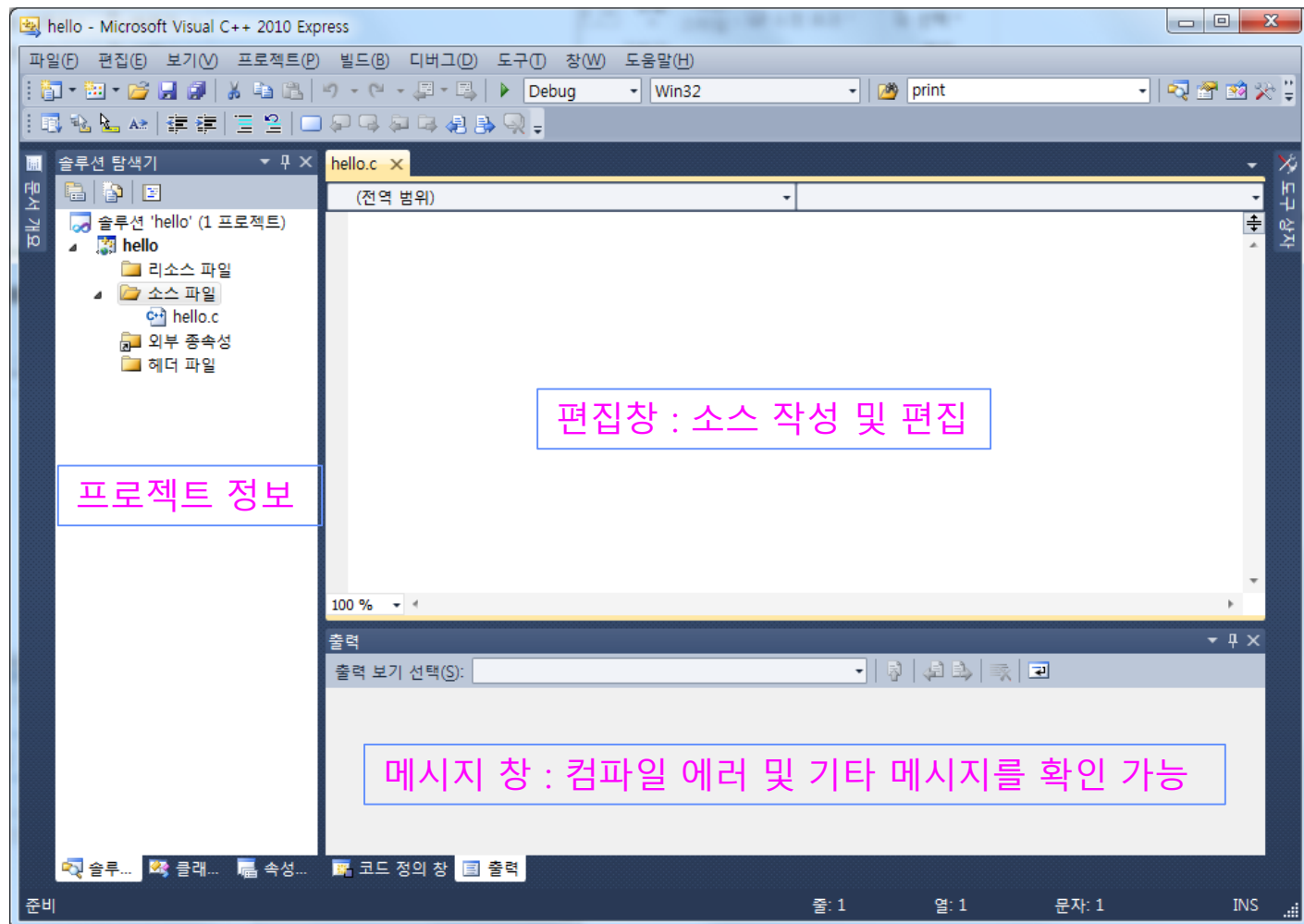
#### • 소스 파일 생성하기

- ✓ 소스파일 우 클릭 → 추가 → 새 항목
- ✓ C++파일 선택 (주의!! 반드시 파일의 확장명을 .c 로 한다)



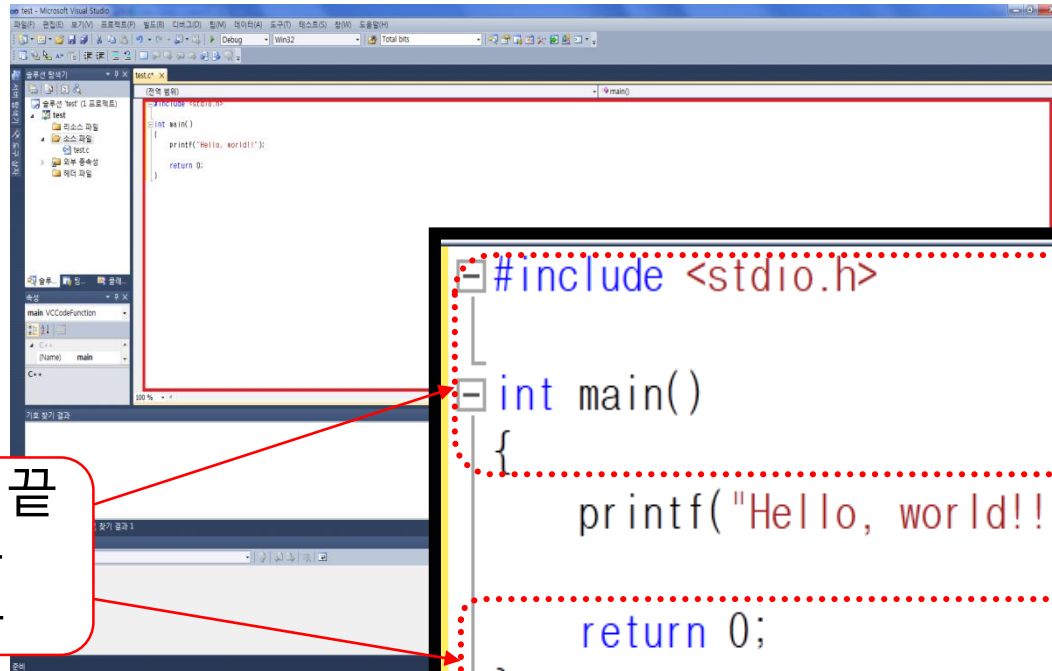
## 2) C 프로그램 작성하기

- 2단계 - 소스코드 작성 (2)
  - VS 창 구성



## 2) C 프로그램 작성하기

- 2단계 - 소스코드 작성 (3)
  - 코드 작성: 편집 창에 코드를 작성한다.



프로그램의 시작과 끝  
부분은 앞으로 계속  
나오니 기억해 두자



## 2) C 프로그램 작성하기

---

- 2단계 - 소스코드 작성 (4)
  - 소스코드 (hello.c)

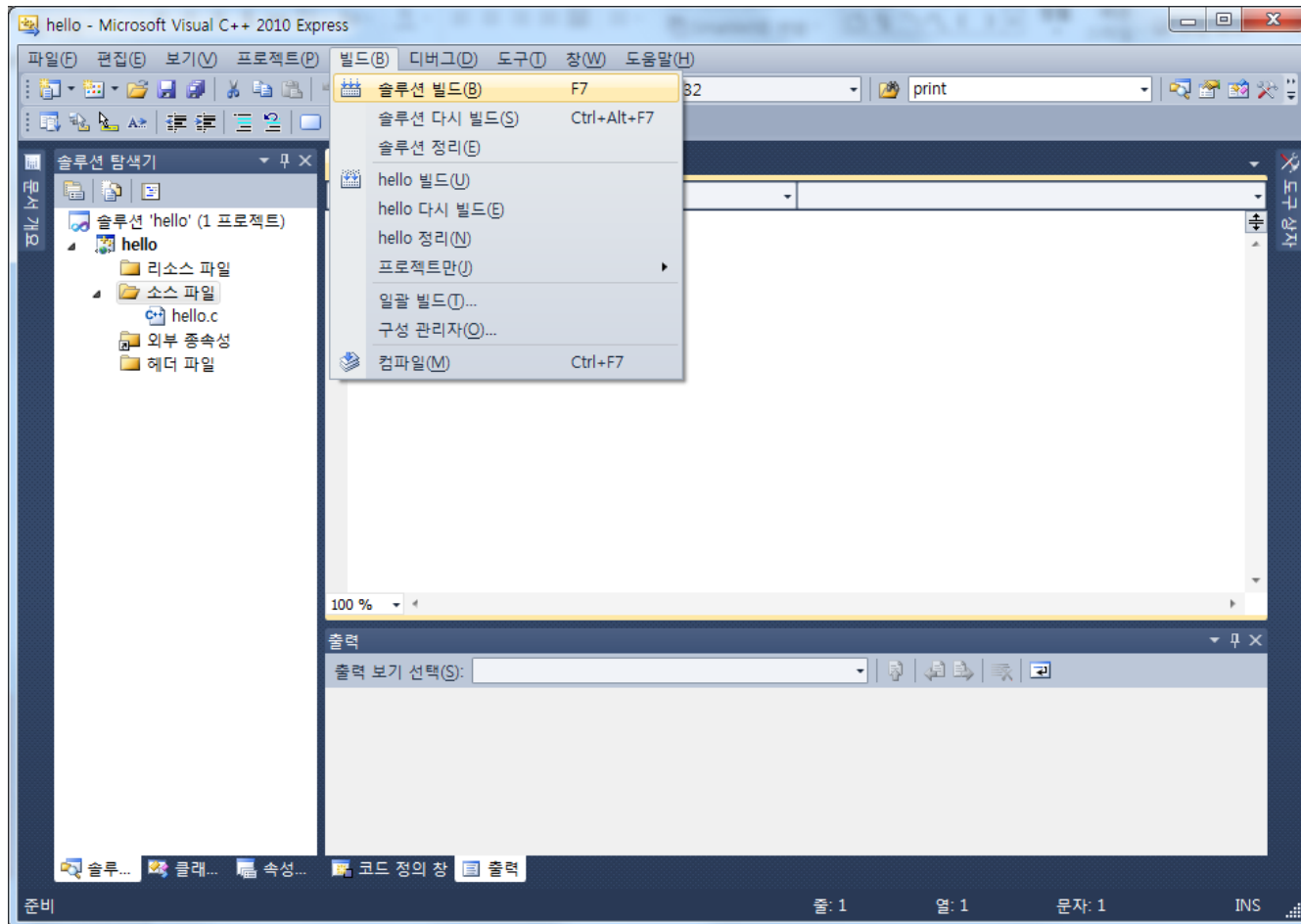
```
#include <stdio.h>

int main()
{
    printf("Hello, World!!");

    return 0;
}
```

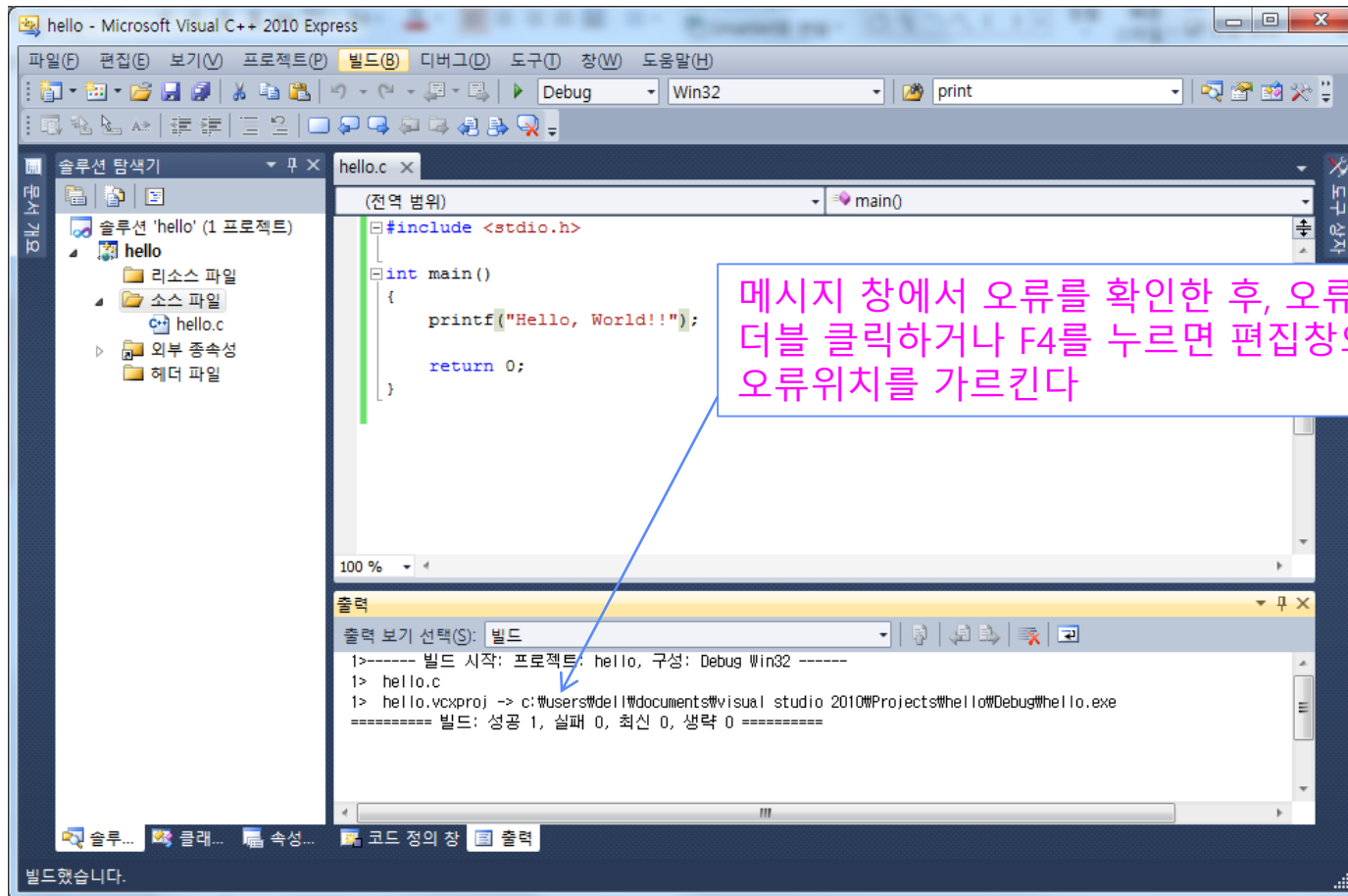
## 2) C 프로그램 작성하기

- 3단계 - 프로그램 빌드 (1)
  - '빌드' 메뉴 → 솔루션 빌드 (단축키: F7)



## 2) C 프로그램 작성하기

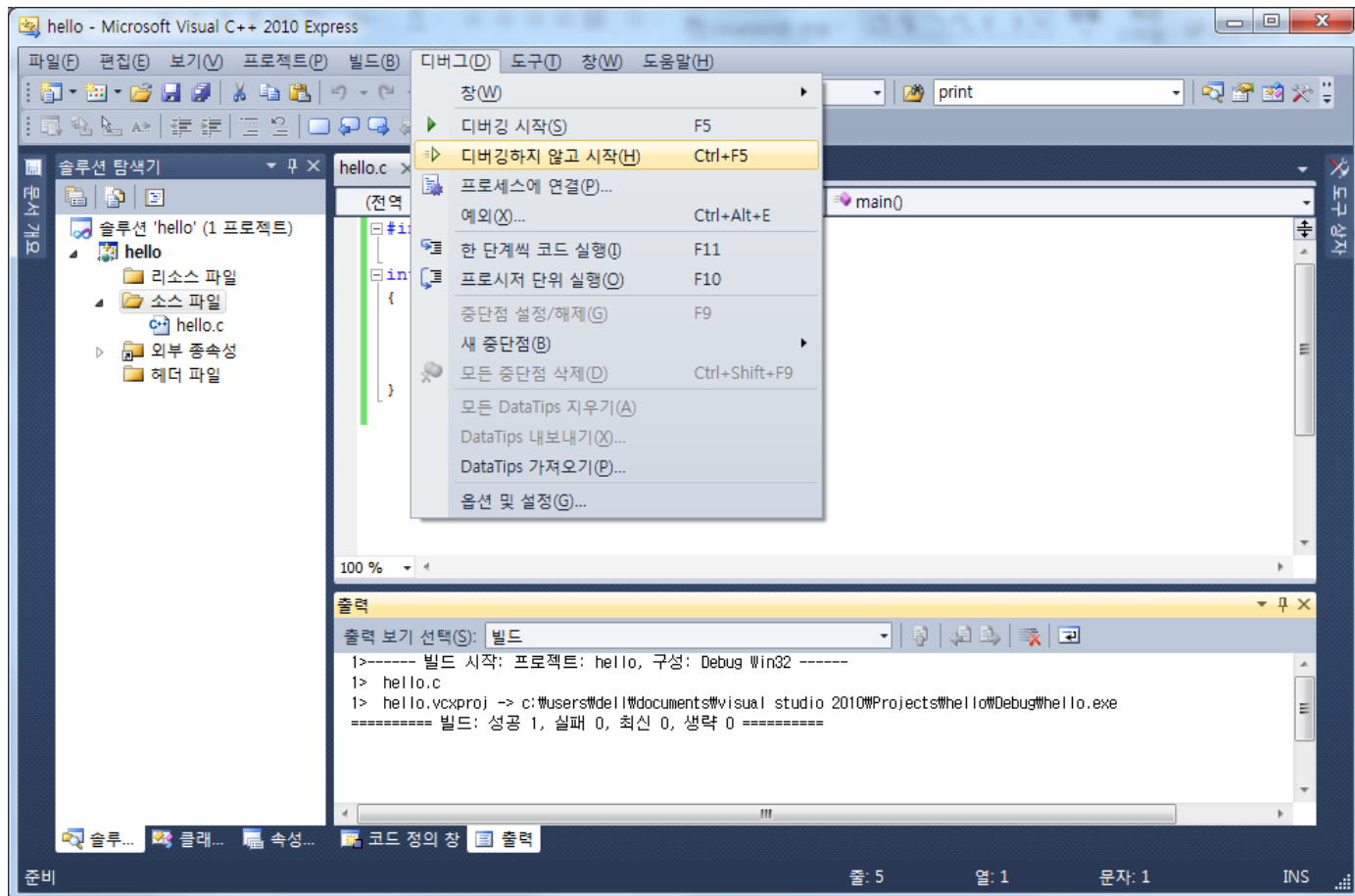
- 3단계 - 프로그램 빌드 (2)
  - 컴파일 및 빌드 결과 확인



## 2) C 프로그램 작성하기

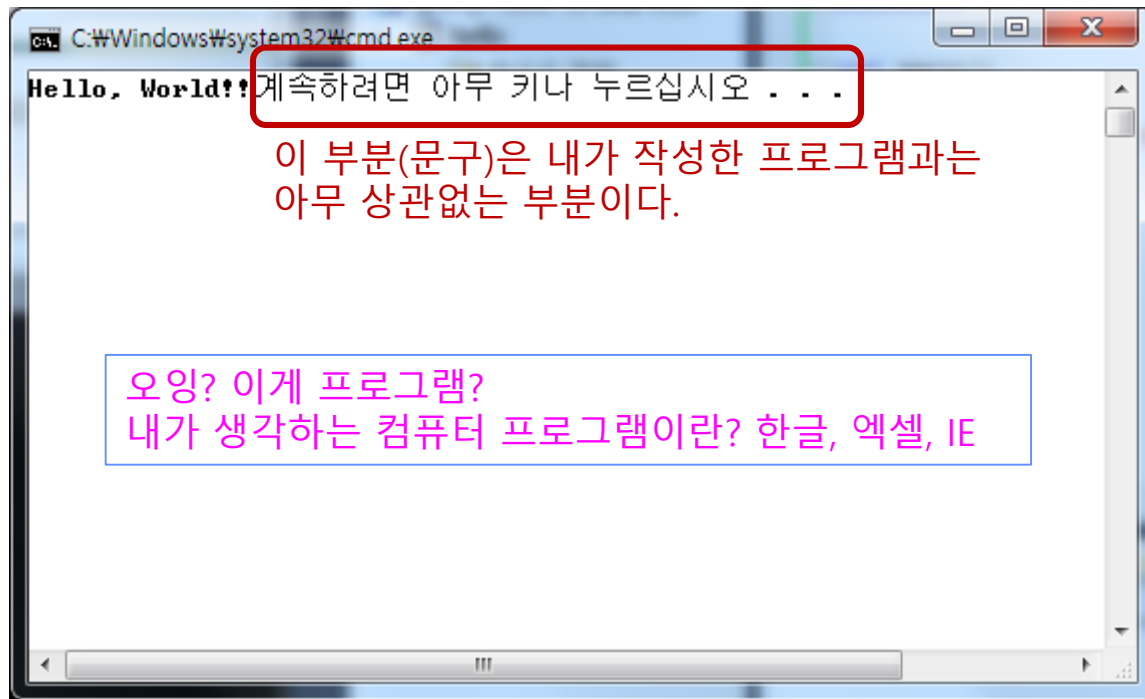
### ■ 4단계 - 프로그램 실행 (1)

- '디버그' 메뉴 → 디버깅하지 않고 시작 (단축키: **Ctrl+F5**)



## 2) C 프로그램 작성하기

- 4단계 - 프로그램 실행 (2)
  - 실행 결과 확인: 프로그램의 실행 결과가 새로운 창에 뜬다



## 2) C 프로그램 작성하기

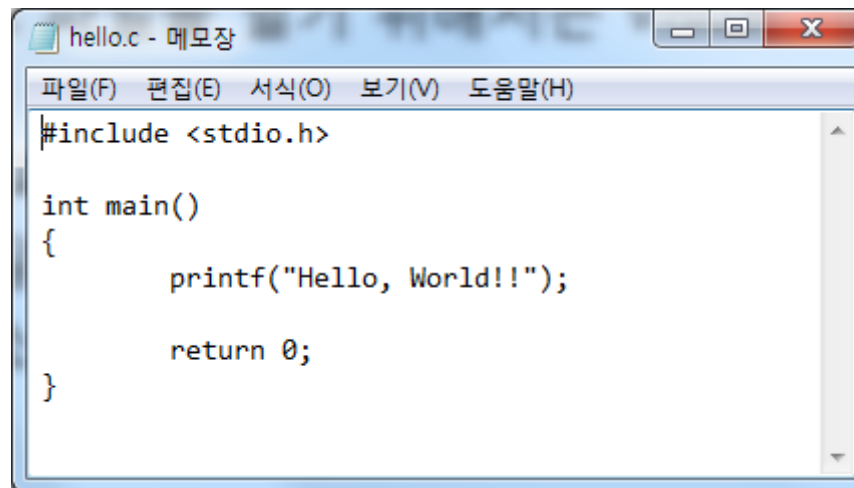
---

- [실습] hello world 프로그램을 VS를 이용하여 작성해보고 실행시켜보자.

## 2) C 프로그램 작성하기

### ■ 소스 파일의 실체

- 소스파일(.c)는 어떤 파일일까?
  - ✓ .hwp 파일을 열기 위해 한글 프로그램이 필요했듯이,
  - ✓ .c 파일을 열기 위해서는 Visual Studio가 필요? NO!!
- 소스파일(.c)을 메모장으로 열어보자
  - ✓ 잘 열리는가? YES
  - ✓ 소스파일(.c)은 텍스트 파일 그 이상도 이하도 아니다.



```
hello.c - 메모장
파일(F)  편집(E)  서식(O)  보기(V)  도움말(H)
#include <stdio.h>

int main()
{
    printf("Hello, World!!");

    return 0;
}
```

# 목차

---

- 1) 프로그래밍 개요
- 2) C 프로그램 작성하기
- 3) 첫 프로그램 분석**
- 4) 출력 기초
- 5) 입력 기초



### 3) 첫 프로그램 분석

- 나의 첫 C 프로그램 (hello.c)
  - 각 라인의 의미

표준 입출력(standard input output)을 포함(include) 하라

main 함수  
(프로그램 시작 지점)

main 함수 시작  
(여는 중괄호)

main 함수의 끝  
(닫는 중괄호)

```
#include <stdio.h>
```

```
int main( )
```

```
{
```

```
    printf("Hello, World!!");
```

```
    return 0;
```

```
}
```

화면에 "." 부분을 출력(print)하라

일단은 그대로 쓰자

프로그램의 각 줄을 하나씩 지워 보고 빌드&실행 하면서  
어떤 결과가 나오는 지 확인해보자

### 3) 첫 프로그램 분석

- 들여쓰기와 주석 : 가독성을 위한 수단으로 문법적 의미는 없음
  - **들여쓰기**: 라인의 포함 여부를 쉽게 파악하기 위함
  - **주석**: 코드에 추가된 프로그램에 대한 설명

```
/* 나의 첫 c 프로그램
Hello world program */

#include <stdio.h>    // 헤더 파일 포함

int main( )          // main 함수 (프로그램 시작 지점)
{                    // main 함수의 시작
•----▶ printf("Hello, world!!");    // 화면에 "..."의 내용을 출력
•----▶ return 0;                    // 0을 반환
}                          // main 함수의 끝
```

들여쓰기와 주석은 선택이 아니라 **필수!!**

### 3) 첫 프로그램 분석

- 들여쓰기: 보통 [tab] 키나 [space] 키를 사용
  - 아래 코드에서 붉은 점이 표시된 두 라인이 main( ) 함수에 속해있음을 알아보기 쉽게 하기 위해 들여쓰기 했음
  - 들여쓰기는 문법적으로 아무런 의미를 가지지 않음
    - ✓ 두 라인이 main( ) 함수에 속해있다는 것을 나타내는 문법적 장치는 **중괄호 쌍**
    - ✓ **[실습]** int main() 이후의 내용을 한 줄에 작성해보고, 컴파일&실행시켜 보자.

```
#include <stdio.h>
int main( )
{
  ●.....▶printf("Hello, world!!");
  ●.....▶return 0;
}
```

### 3) 첫 프로그램 분석

- 주석

- `/* ... */` : `/*`과 `*/` 사이의 내용을 모두 주석으로 처리
- `//` : `//` 이 후부터 해당 라인 끝까지의 내용을 주석 처리

```
/* 나의 첫 c 프로그램
Hello world program */

#include <stdio.h>    // 헤더 파일 포함

int main( )          // main 함수 (프로그램 시작 지점)
{                    // main 함수의 시작
    printf("Hello, world!!"); // 화면에 "... "의 내용을 출력
    return 0;         // 0을 반환
}                    // main 함수의 끝
```

### 3) 첫 프로그램 분석

- 오류의 종류와 디버깅(1)

- 버그(bug) : 프로그램에 존재하는 오류
- 디버깅(debugging) : 오류를 고치는 행위

- 컴파일 오류: 문법적 오류

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

⇒ 세미콜론 누락  
⇒ return 철자 틀림

- ✓ 컴파일러가 문법을 검사하여 오류가 있으면 알려줌
- ✓ [실습] 위 코드를 컴파일해보자.

- 컴파일 경고: 오류는 아니지만, 오류의 가능성이 있는 부분

### 3) 첫 프로그램 분석

- 오류의 종류와 디버깅(2)

- 런타임 오류(runtime error) 또는 실행 오류: 프로그램의 결과가 의도와 다르거나 비정상적으로 종료되는 경우

```
#include <stdio.h>
int main( ) {
    printf("Jello, World!!")  ⇒ H를 J로 잘못 타이핑
    return 0;
}
```

- ✓ 위 예는 매우 단순한 오류이나, 일반적으로 런타임 오류는 찾기도 어렵고 해결하기도 어려움
- ✓ 보통 '디버깅'이라 함은 런타임 오류를 해결하는 것을 지칭
- 대부분의 개발도구에는 런타임 오류를 해결에 필요한 부가 기능 제공 → VS의 디버깅 메뉴 (간단한 사용법은 교재 부록 참조)

# 목차

---

- 1) 프로그래밍 개요
- 2) C 프로그램 작성하기
- 3) 첫 프로그램 분석
- 4) **출력 기초**
- 5) 입력 기초

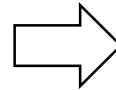
## 4) 출력 기초

### ■ 화면에 내용 출력하기

- printf() 함수는 큰 따옴표의 시작과 끝 사이에 있는 내용 출력
- 큰 따옴표 밖의 공백은 출력에 전혀 영향 없고, 내부 공백만 출력에 영향을 줌

소스 코드

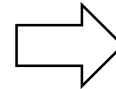
```
printf( "Hello,   world!!" );
```



실행 화면

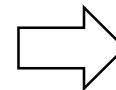
```
Hello,   world!!
```

```
printf( "  I'm a student." );
```



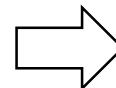
```
I'm a student.
```

```
printf(   "100"   ) ;
```



```
100
```

```
printf(   "!@# $"  );
```



```
!@#$
```



## 4) 출력 기초

- **printf** 함수를 여러 번 사용하면?
  - C 언어의 문장은 기본적으로 위에서부터 순차적으로 수행

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    printf("한국대학교");
```

```
    printf("신입생");
```

```
    printf("홍길동");
```

```
    return 0;
```

```
}
```

순차  
실행

실행 화면

한국대학교신입생홍길동

- 심화 1: 위 내용을 띄어 쓰기를 넣어서 출력해보자.
- 심화 2: 위 내용을 printf() 함수 하나만 사용해서 출력해보자

## 4) 출력 기초

- 앞의 결과를 오른쪽과 같이 3개의 줄에 출력하려면?

- 개행 문자 (\n) 사용

- ✓ 참고) 역슬래시 기호(\)는  
한글 폰트에서는 원화 기호(₩)로 보임

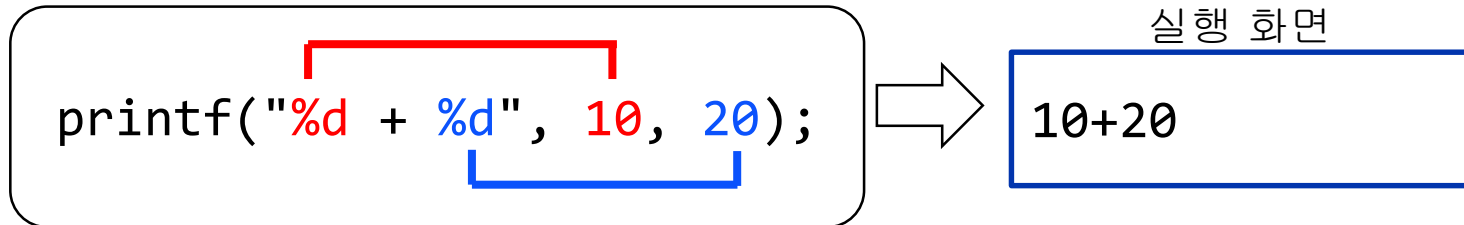
실행 화면

```
한국대학교  
신입생  
홍길동
```

```
printf("한국대학교\n");
```

## 4) 출력 기초

### ▪ 정수 값 출력



\* %d 는 정수형 값을 출력하는 서식

- '%d'는 해당 위치에 정수를 출력하라는 의미 (서식 지정자)
- 출력할 값은 두 번째 큰 따옴표 뒤에 명시
- 하나의 printf 함수에 여러 개의 서식이 삽입될 수 있으며, 문장 앞쪽의 서식부터 차례로 대응

## 4) 출력 기초

---

- [예제 1.1] – 실습

- 다음 프로그램의 결과를 예측해보고, 프로그램을 작성하여 결과를 확인해 보자.

```
int main()  
{  
    printf("100");  
    printf("\n");  
    printf("%d", 100);  
    printf("\n");  
    return 0;  
}
```

```
int main( )  
{  
    printf("100+200");  
    printf("\n");  
    printf("%d", 100+200);  
    printf("\n");  
    return 0;  
}
```

## 4) 출력 기초

- 변수에 저장된 값 출력

- 변수에 저장된 값도 동일한 방식으로 화면에 출력
- 변수에 대한 자세한 내용은 2장에서 학습

```
int main()
{
    int c;          // 변수 c 선언

    c = 10+20;      // c 에 10+20 의 결과, 즉 30 저장

    printf("c=10+20 출력:"); // "... " 내용 출력
    printf("%d", c);         // 변수 c에 저장된 값 출력
    return 0;
}
```

# 목차

---

- 1) 프로그래밍 개요
- 2) C 프로그램 작성하기
- 3) 첫 프로그램 분석
- 4) 출력 기초
- 5) **입력 기초**

## 5) 입력 기초

- 키보드를 이용하여 값 입력 받기
  - scanf( ) 함수는 큰 따옴표 "..." 안에 있는 서식 지정자에 따라 값을 입력 받음

```
int main()
{
    int a;                // 변수 a 선언

    printf("정수를 입력하시오: "); // 안내문 출력
    scanf("%d", &a);        // 정수 값 입력 받아
                           // 변수 a에 저장
    printf("입력된 값: %d", a); // 변수 a의 값 출력

    return 0;
}
```

## 5) 입력 기초

---

- 앞의 프로그램을 작성하여 실행하면?

- 화면에 "정수를 입력하시오: "를 출력한 후,
- 프로그램은 키보드로 값이 입력되길 기다림
- 아무 정수나 입력하고 [Enter] 키를 치면, 나머지 부분이 실행됨

실행 화면

정수를 입력하시오: **15**  
입력된 값: 15

### ※ 참고


- VS에서 scanf\_s( ) 함수를 사용하라고 하면서 컴파일 오류가 발생하는 경우 아래 문장을 소스코드에 추가

```
#pragma warning(disable:4996)
```



## 5) 입력 기초

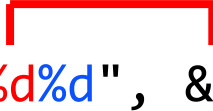
- scanf() 함수 사용법



```
scanf("%d", &a);
```

변수 앞에 &를 반드시 붙이자

- 키보드로 입력된 정수 값을 변수 a에 저장
- %d: 입력되는 값을 정수로 해석하라는 의미의 서식 지정자
- printf 함수에서와 달리, 변수 a 앞에 & 기호를 붙여야 한다.
  - ✓ 이유는 나중에
- 여러 개의 정수 입력 받기



```
scanf("%d%d", &a, &b);
```

## 5) 입력 기초

- [예제 1.2] – 실습 (변수와 printf, scanf를 여러 번 사용)

```
int main()
{
    int id;           // 변수 선언
    int age;

    printf("학번 입력: ");
    scanf("%d", &id);
    printf("입력 학번: %d\n", id);

    printf("나이 입력: ");
    scanf("%d", &age);
    printf("입력 나이: %d\n", age);

    return 0;
}
```

실행 화면

학번	입력:	<u>1601111</u>
입력	학번:	1601111
나이	입력:	<u>20</u>
입력	나이:	20

// 안내문 출력  
// 값 입력  
// 입력된 값 출력

## 5) 입력 기초

---

- [예제 1.3] – 실습

- 학번과 나이를 scanf 함수를 한번만 사용하여 입력 받아 다음과 같은 실행 결과가 나오도록 프로그램을 수정하십시오.

실행 화면

```
학번과 나이 입력: 1601111 20  
입력 학번과 나이: 1601111 20
```

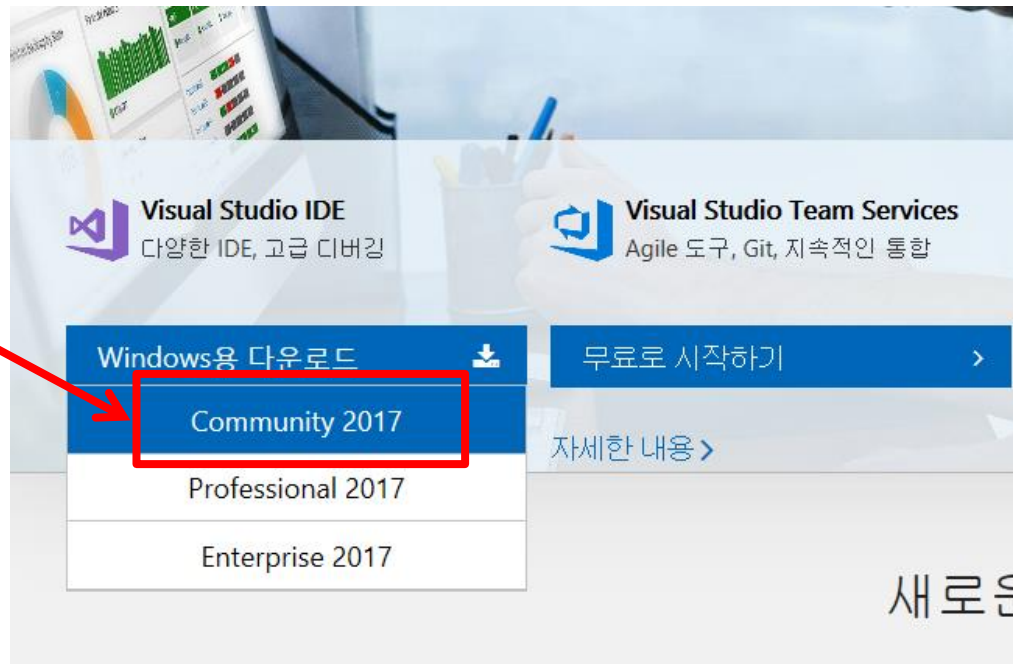
# Visual Studio 설치 방법

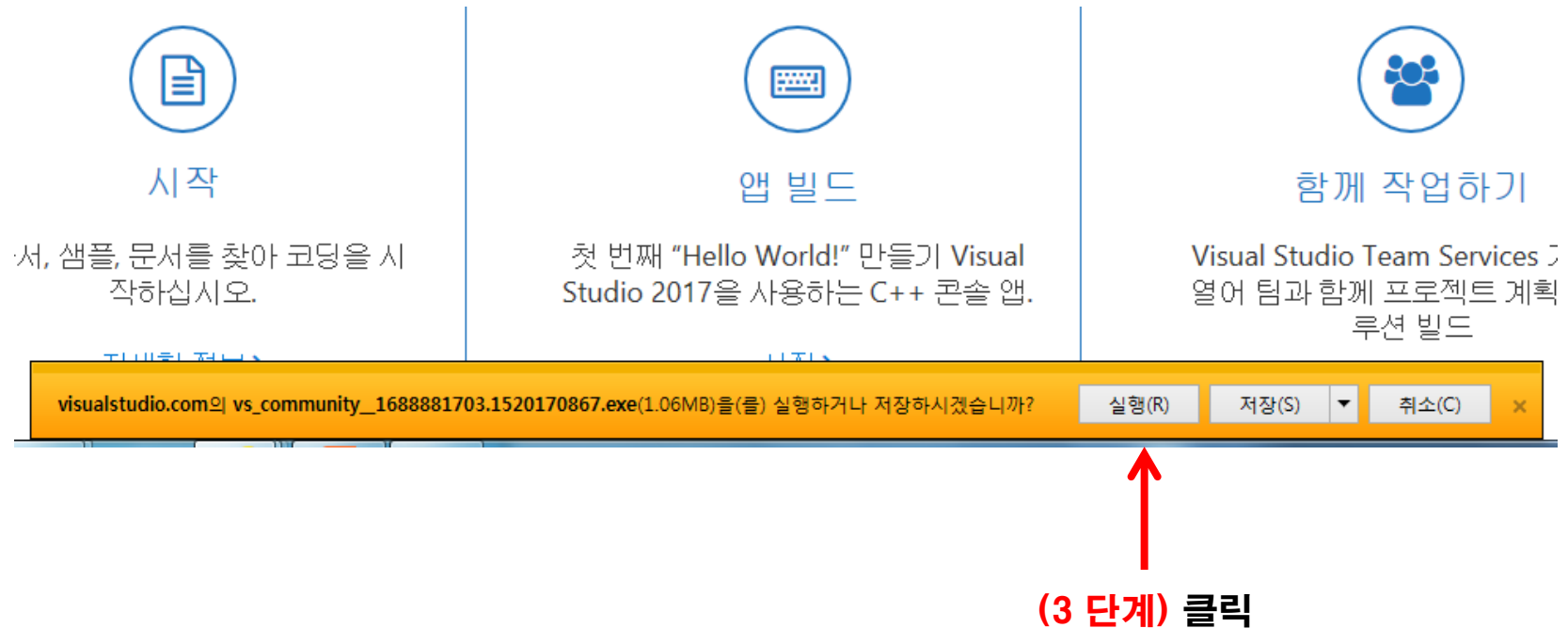
무료로 visual studio를 설치 할 수 있습니다. (실습시간에 이용한 버전은 2012버전 입니다.)

(1 단계) <https://www.visualstudio.com/ko>

위 링크로 접속하여 windows용 **community 2017**을 설치한다. (Windows에 C++개발환경을 선택)

(2 단계)





### 이용방법 (VS2017의 프로젝트 생성은 교재 참고)

2. visual studio를 실행 후 파일-> 새로 만들기-> 프로젝트-> 콘솔응용프로그램 선택-> 저장위치 선택-> 빈 프로젝트 체크
3. 솔루션탐색기에서 소스파일 우클릭-> 추가-> 새 항목을 통해 새 파일을 만들어준다.(ctrl+shift+a)

Visual C++ -> 일반 -> 빈프로젝트