

4주차 1차시

## 프로그래밍 언어(1)

인간과 컴퓨터의 대화\_진화하는 소통·진화하는 컴퓨터

- 1 프로그래밍 언어의 개요
- 2 프로그래밍 언어의 실행 과정



## 프로그래밍 언어(1)

### ◆ 학습목표

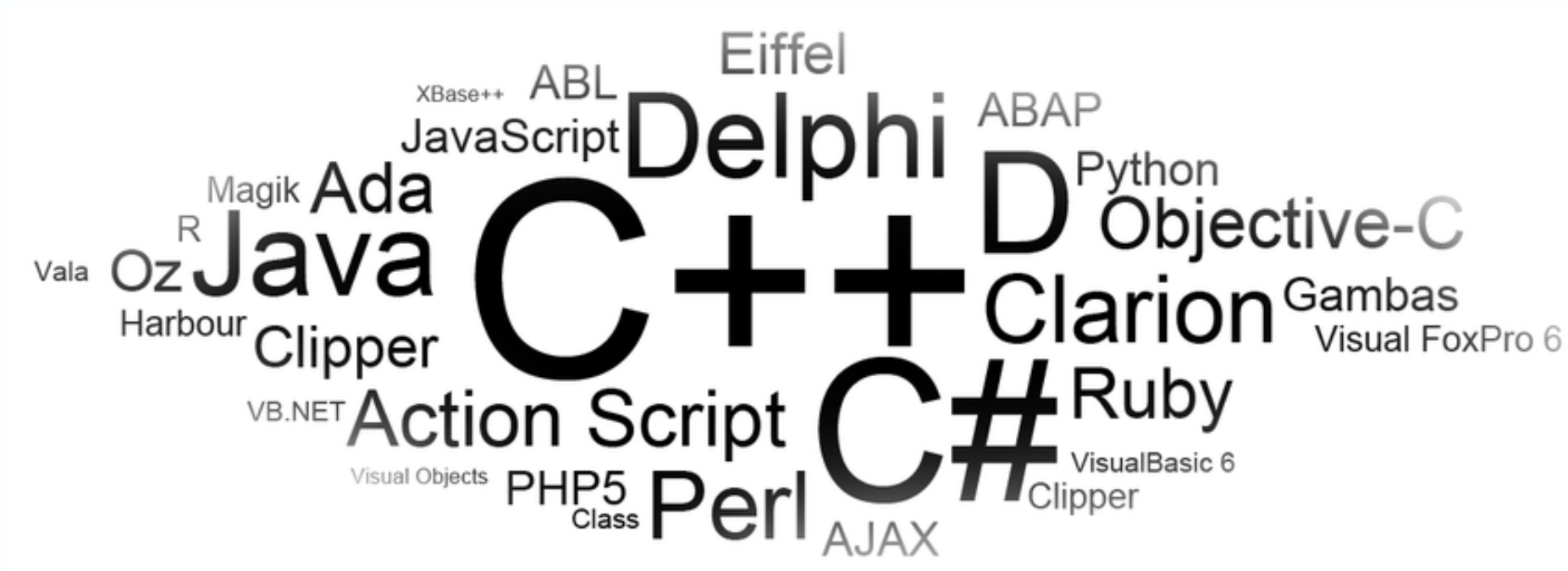
- 프로그래밍 언어의 기본 개념과 특징을 알아본다.
- 프로그래밍 언어의 발전 과정과 기술 동향을 알아본다.
- 프로그래밍 언어의 종류와 구현 원리를 알아본다.
- 프로그래밍 언어의 실행 과정을 알아본다.

# 1. 프로그래밍 언어의 개요

## 1) 프로그래밍 언어의 개념

### ◆ 프로그래밍 언어

- 인간이 컴퓨터와 의사소통할 수 있도록 컴퓨터에 내리는 명령으로 프로그램을 처리하도록 기술한 언어를 말함



# 1. 프로그래밍 언어의 개요

## 2) 저급 언어와 고급 언어

### ◆ 저급 언어

- 컴퓨터 내부 표현에 가까운 언어로 기계어와 어셈블리어로 구분
- 기계어 : 0과 1로 된 2진수
- 어셈블리어 : 기계어 명령을 알기 쉬운 기호로 표시한 것

연산 코드 기호	의미
LOAD X	$CON(X) \rightarrow R$
STORE X	$R \rightarrow CON(X)$
CLEAR X	$0 \rightarrow CON(X)$
ADD X	$R + CON(X) \rightarrow R$
INCREMENT X	$CON(X) + 1 \rightarrow R$
SUBTRACT X	$R - CON(X) \rightarrow R$
DECREMENT X	$CON(X) - 1 \rightarrow CON(X)$
COMPARE X	if $CON(X) R$ then $GT = 1(on)$

어셈블리 언어로 표현된 명령어들의 예

# 1. 프로그래밍 언어의 개요

## 2) 저급 언어와 고급 언어

### ◆ 고급 언어

- 고급 언어의 특징
  - 일상 언어에서 사용하는 표현을 그대로 가져다 쓸 수 있음
  - 사용자가 기억 장소의 주소를 일일이 기억할 필요가 없음
  - 하나의 명령어로 다수의 연산을 실행
- 고급 언어로 작성한 프로그램을 실행하는 과정

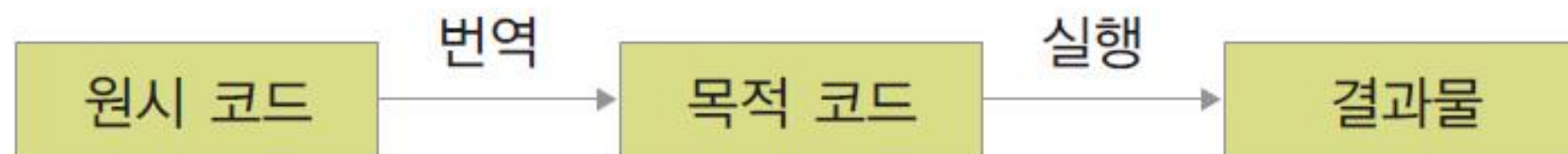


그림 4-1 프로그램 작성과 실행 과정

- 대표적인 고급 언어
  - C, 포트란, 코볼, 파스칼, C++, 자바, 스몰토크 등

# 1. 프로그래밍 언어의 개요

## 3) 프로그래밍 언어

### ◆ 컴파일러 언어와 인터프리터 언어

- 고급언어를 기계어로 번역해주는 도구, 또는 논리적 장치

특징 \ 방식	컴파일러 방식	인터프리터 방식
번역방법	프로그램 전체 번역	실행되는 행 단위로 번역
장점	한번 번역하면 빠른 시간 내에 전체를 실행가능	큰 기억 장치가 필요하지 않으며 번역 과정이 비교적 간단함
단점	프로그램의 일부를 수정하는 경우에도 전체를 컴파일해야 함	반복문이 많은 경우 매 반복 때마다 번역해야 함
결과물	목적 언어로 된 프로그램	실행의 결과
적용언어	FORTTRAN, Pascal, COBOL, Ada, C 등	Lisp, SNOBOL 4, Prolog 등

컴파일러 방식과 인터프리터 방식의 특성 비교

# 1. 프로그래밍 언어의 개요

## 4) 프로그래밍 언어의 발전

### ◆ 1950년대 언어

- 포트란 개발 → 프로그래밍 언어 발전의 이정표가 됨

### ◆ 1960년대 언어

- 과학기술용으로 개발된 포트란을 더욱 발전시킨 고급 언어와 사무처리용 고급 언어 출현
- 대표적인 사무처리용 언어 '코볼'

### ◆ 1970년대 언어

- C언어와 파스칼이 개발됨

# 1. 프로그래밍 언어의 개요

## 4) 프로그래밍 언어의 발전

### ◆ 1980년대 언어

- 단말 시스템을 이용한 분산 처리 개념이 확산
- 학생들과 컴퓨터 초보자에게 적합한 교육용 언어가 요구 → 베이직 언어 등장

### ◆ 1990년대 언어

- 1990년대에는 객체 지향 언어가 본격적으로 등장
- C++, 자바, 비주얼 베이직 등의 객체 지향 언어가 새로 등장



# 1. 프로그래밍 언어의 개요

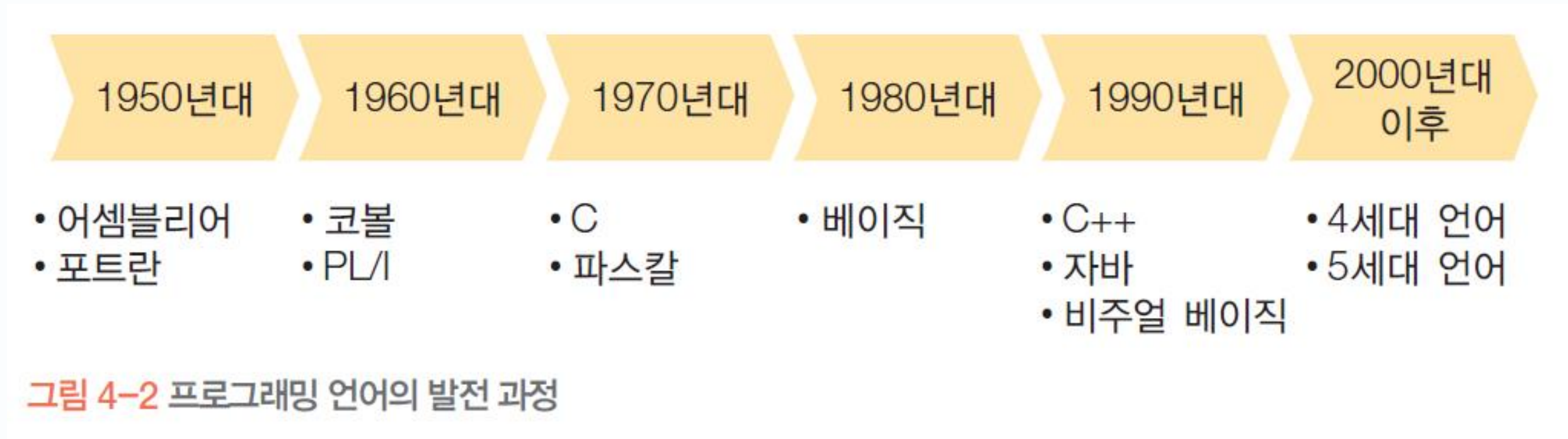
## 4) 프로그래밍 언어의 발전

### ◆ 2000년대 언어

- 파워빌더, 델파이, 각종 쿼리 전용 언어 등 소위 4세대라 불리는 언어 등장
- 소프트웨어 컴포넌트 기술 발전
- 객체 지향 기술과 웹의 결합을 통해 다양한 정보를 제공하는 기법도 발전
- 최근에는 5세대 언어라 불리는 인공지능 기능을 이용해 자연 언어로 직접 처리하는 기법에 대한 연구가 진행됨

# 1. 프로그래밍 언어의 개요

## 4) 프로그래밍 언어의 발전



# 1. 프로그래밍 언어의 개요

## 5) 주요 프로그래밍 언어별 특징

### ◆ 코볼

표 4-1 코볼의 장단점

장점	<ul style="list-style-type: none"><li>· 컴퓨터의 내부적인 특성과 별개로 설계되어 코볼 컴파일러만 있으면 컴퓨터 기종에 관계없이 사용할 수 있다.</li><li>· 파일의 순차 처리와 비순차 처리를 모두 할 수 있어 다른 프로그래밍 언어에 비해 파일 처리 기능이 강력하다.</li><li>· 작성이 쉽고 이해하기 쉽다.</li></ul>
단점	<ul style="list-style-type: none"><li>· 컴파일러가 많은 항목을 포함하고 있어 주기억장치 용량을 많이 차지한다.</li><li>· 프로그램 작성량이 많고 길어서 전체적으로 간결하지 못하다.</li></ul>

표 4-2 코볼 프로그램의 구성

디비전	설명	기술 내용
IDENTIFICATION	프로그램의 내용을 파악하는 식별 디비전	프로그램 이름, 작성자, 작성 일자 등
ENVIRONMENT	프로그램의 처리에 관계되는 환경 디비전	컴퓨터 종류, 입출력 파일 및 장치
DATA	데이터 처리를 위한 기억 장소 디비전	기억 장소 형식, 성격, 크기, 내용 등
PROCEDURE	처리할 명령에 관한 구체적 기술 디비전	처리 순서에 따른 명령문 실행의 기술

# 1. 프로그래밍 언어의 개요

## 5) 주요 프로그래밍 언어별 특징

### ◆ 파스칼

- 복합문 begin-end, 조건문 if-then-else, 반복문 while-do와 같은 제어 구조가 있어 구조적 프로그래밍에 적합

```
procedure squareroots (input,output)
var
x : real;
begin
repeat
read(x);
if x ≥ 0
then write(sqrt(x))
else write ('argument error')
until x = 0
end
```

# 1. 프로그래밍 언어의 개요

## 5) 주요 프로그래밍 언어별 특징

### ◆ 비주얼 베이직

표 4-3 비주얼 베이직의 장단점

장점	<ul style="list-style-type: none"><li>· 쉽고 간편하게 작성할 수 있어 초보자나 학생들의 교육용으로 사용하기 좋다.</li><li>· 한글 지원이 우수하다.</li><li>· <u>마이크로소프트</u>에서 제공하는 각종 툴을 편하게 이용할 수 있다.</li></ul>
단점	<ul style="list-style-type: none"><li>· 객체 지향 기능이 C#이나 자바 등에 비해 약하다.</li></ul>

# 1. 프로그래밍 언어의 개요

## 5) 주요 프로그래밍 언어별 특징

### ◆ 비주얼 베이직

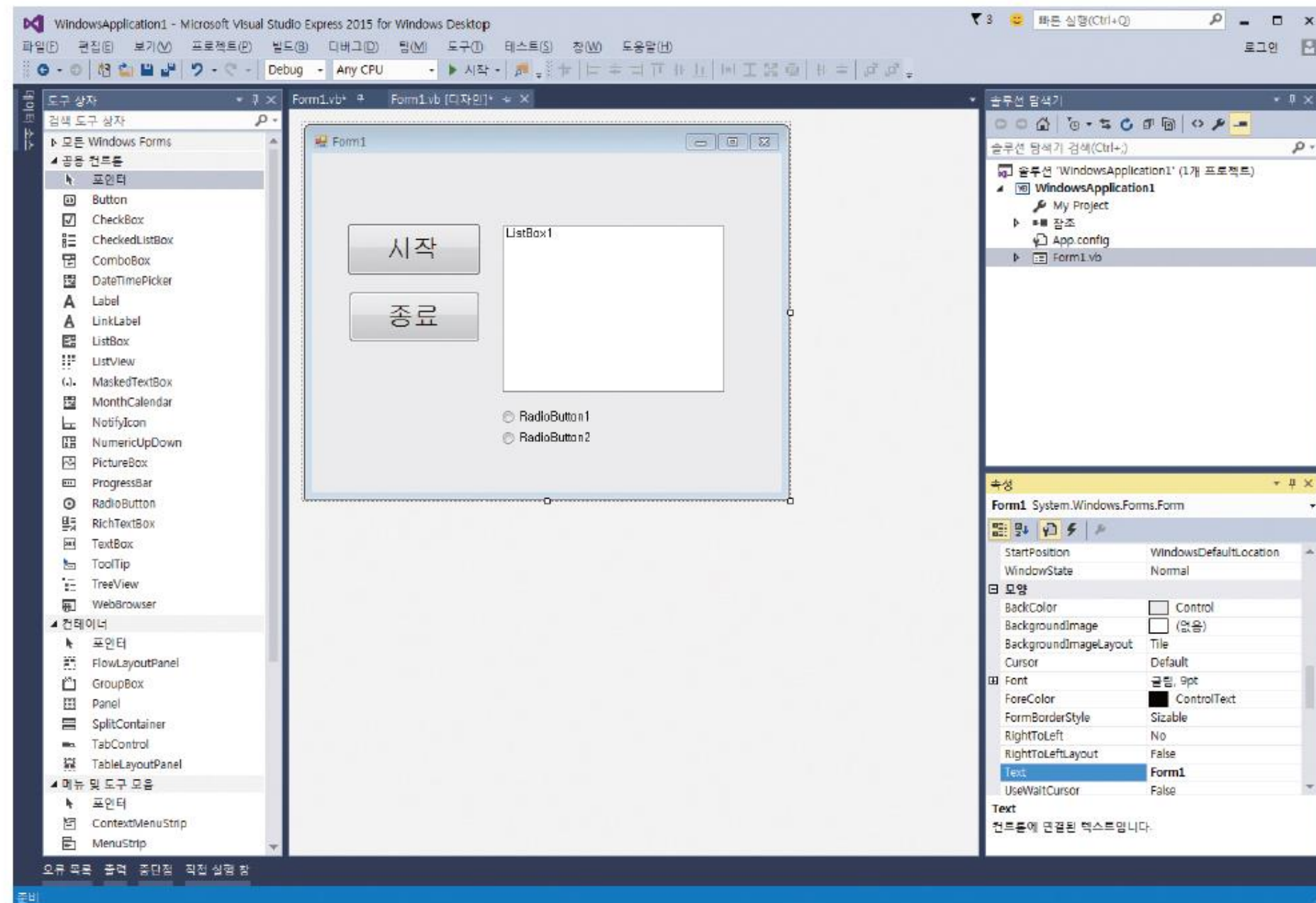


그림 4-3 마이크로소프트 비주얼 스튜디오 2015의 비주얼 베이직 프로그램 화면



# 1. 프로그래밍 언어의 개요

## 5) 주요 프로그래밍 언어별 특징

### ◆ C언어 계열(C,C++, C#)

표 4-4 C언어 계열의 장단점

장점	<ul style="list-style-type: none"><li>· 어셈블리어 같은 저급 언어와 유사한 기능을 포함한다.</li><li>· 구조적 프로그래밍 기능이 있어 프로그램을 읽고 작성하기 쉽다.</li><li>· 프로그램의 융통성과 이식성이 상대적으로 뛰어나다.</li><li>· 기존에 C언어로 개발한 프로그램을 거의 수정하지 않고도 C++로 확장할 수 있어 대부분의 운영체제에서 바로 쓸 수 있다.</li><li>· 전 세계 수많은 C 프로그래머가 자연스럽게 C++ 프로그래머로 전환할 수 있어 전문 인력이 부족해지는 문제를 해결할 수 있다.</li></ul>
단점	<ul style="list-style-type: none"><li>· C는 객체 지향 개념이 없다.</li><li>· C++는 방대하고 복잡하여 안정성이 떨어진다. C언어와 호환성이 주요한 특징이므로 새로운 기능을 추가하는 데 한계가 있다.</li><li>· C#은 자바 사용자 층에 비해 사용자 층이 아직까지 활성화되지 못했다.</li></ul>

# 1. 프로그래밍 언어의 개요

## 5) 주요 프로그래밍 언어별 특징

### ◆ 자바

- C++의 강력함을 제공하면서도 규모는 더 작고 안전성은 강화된 언어
- 웹 환경에 적합하다는 것이 큰 장점
- 월드 와이드 웹의 보급 확대와 보조를 맞춰 발전



## 2. 프로그래밍 언어의 실행 과정

### 1) 사용자 요구 사항 분석과 프로그램 설계

#### ◆ 자바

- 사용자의 필요를 파악하고 프로그램을 통해 해결할 문제가 무엇인지 확인하는 단계

#### ◆ 프로그램 설계

- 실제 코딩을 시작할 때 사용할 논리를 프로그래머가 대략 그려내는 단계
- 알고리즘 설계라고도 함
- 알고리즘의 특성
  - 알고리즘 명령을 수행하면 유한한 횟수를 거친 후 종료해야 한다.
  - 알고리즘의 각 단계와 명령은 명확하게 정의되어야 한다.
  - 알고리즘은 데이터 입력이 0 또는 그 이상이어야 한다.
  - 알고리즘은 한 가지 이상의 결과를 출력한다.
  - 알고리즘은 효과적이어야 한다. 이는 유한한 시간 내에 정확히 수행할 수 있을 정도로 단순해야 함을 의미한다.

## 2. 프로그래밍 언어의 실행 과정

### 1) 사용자 요구 사항 분석과 프로그램 설계

#### ◆ 프로그램 설계

- 프로그램 제어 흐름 유형
  - 순차 구조 : 프로그램 코드 순서대로 실행
  - 선택 구조 : 프로그램이 다음에 무엇을 해야 하는지를 결정하는 분기 구조
  - 반복 구조 : 조건이 만족하지 않을 때까지 계속 반복

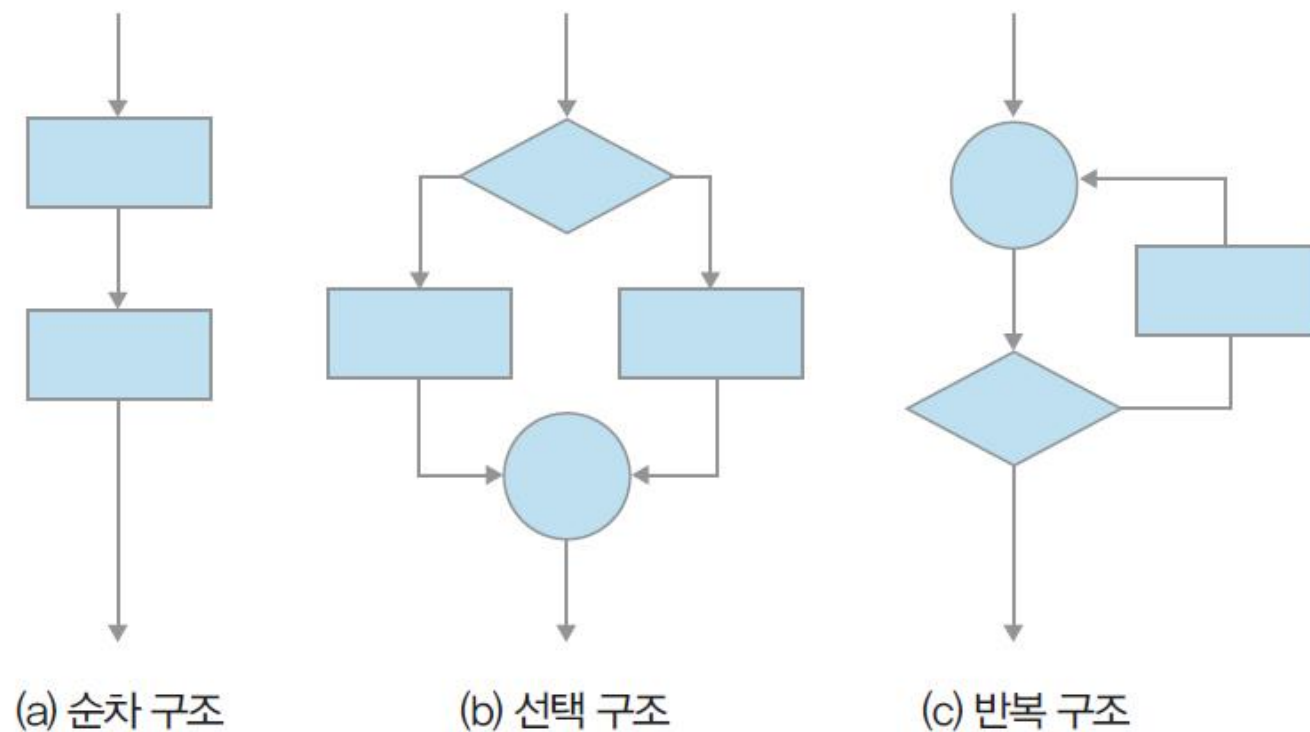


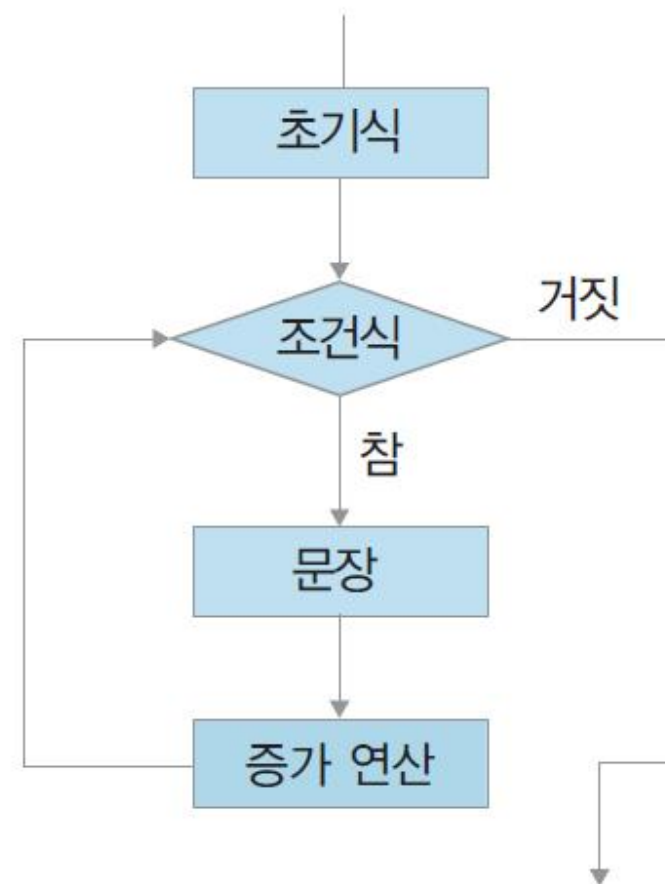
그림 4-4 프로그램 제어 흐름의 유형

## 2. 프로그래밍 언어의 실행 과정

### 1) 사용자 요구 사항 분석과 프로그램 설계

#### ◆ 프로그램 설계

- 반복 구조
  - For문



```

for (초기식; 조건식; 증가 연산) {
    문장;
}
  
```

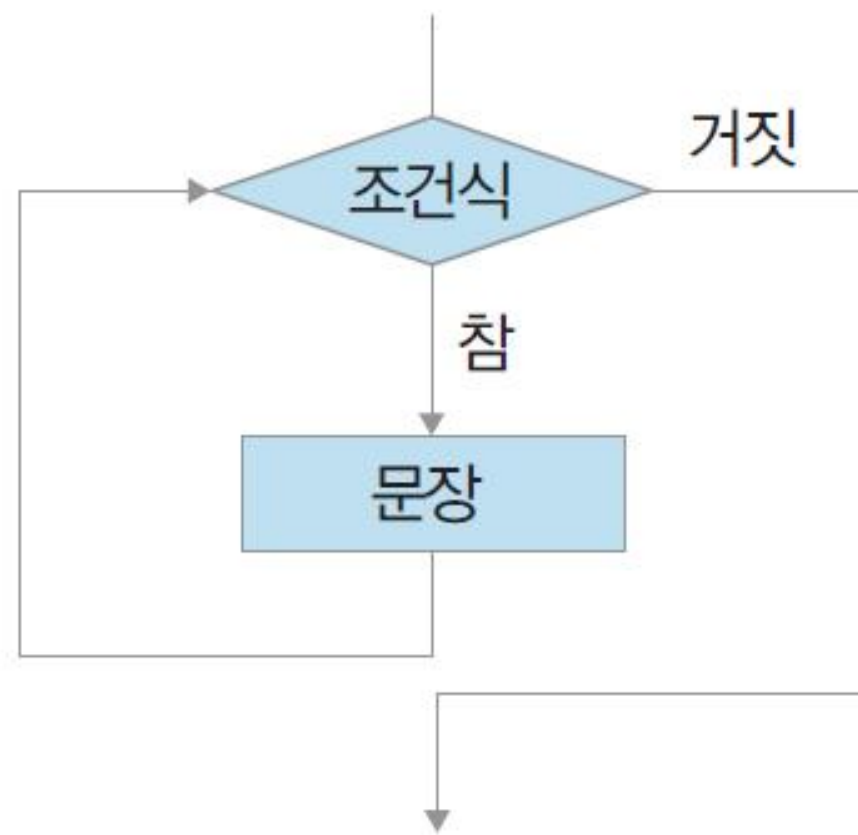
그림 4-5 for 문의 순서도와 명령 형태

## 2. 프로그래밍 언어의 실행 과정

### 1) 사용자 요구 사항 분석과 프로그램 설계

#### ◆ 프로그램 설계

- 반복 구조
  - while문



```
while(조건식) {  
    문장;  
}
```

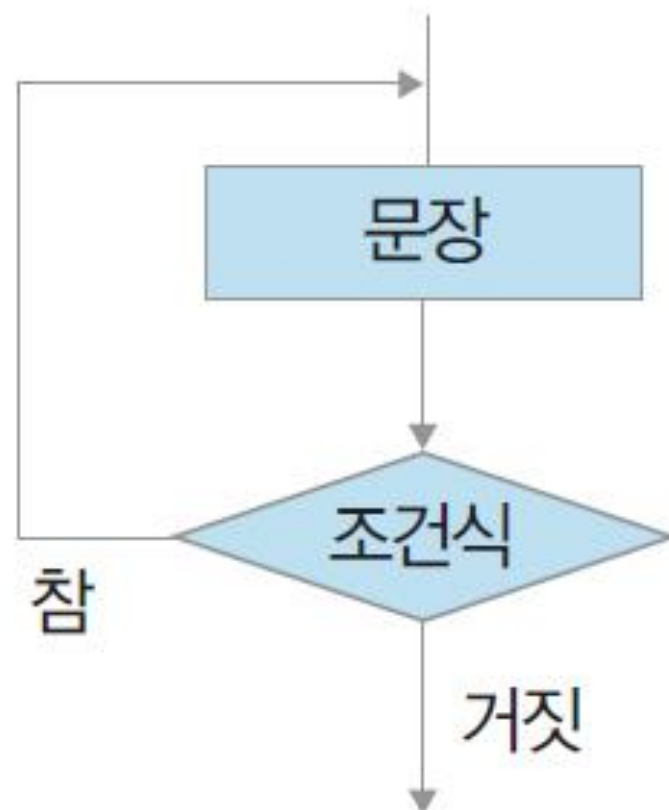
그림 4-6 while 문의 순서도와 명령 형태

## 2. 프로그래밍 언어의 실행 과정

### 1) 사용자 요구 사항 분석과 프로그램 설계

#### ◆ 프로그램 설계

- 반복 구조
  - do-while문



```
do {  
    문장;  
} while(조건식);
```

그림 4-7 do-while 문의 순서도와 명령 형태

## 2. 프로그래밍 언어의 실행 과정

### 2) 코딩 및 컴파일

- ◆ 코딩 : 프로그래밍 언어로 프로그램을 작성하는 단계
- ◆ 컴파일 : 고급 언어로 작성된 명령문을 기계어로 바꾸는 단계
- ◆ 컴파일러를 이용한 방식
  - 프로그램 전체를 한번에 기계어로 번역하는 방식
  - C언어, 코볼, 포트란, 파스칼 등의 언어에서 사용

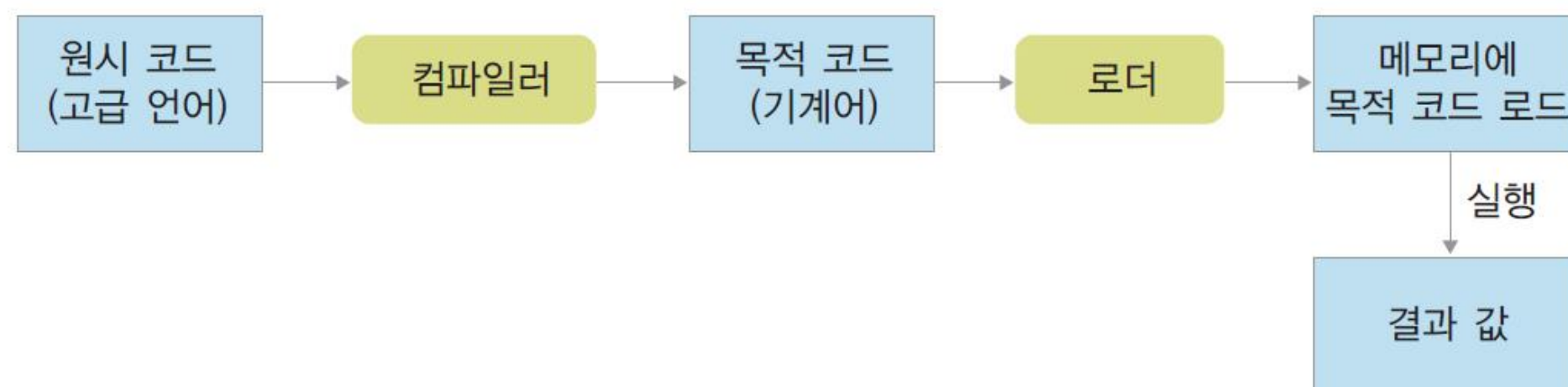


그림 4-8 컴파일러를 이용한 방식

## 2. 프로그래밍 언어의 실행 과정

### 2) 코딩 및 컴파일

#### ◆ 인터프리터를 이용한 방식

- 프로그램을 한 행씩 읽어 번역과 실행을 동시에 하는 방식
- 베이직 등의 언어에서 사용

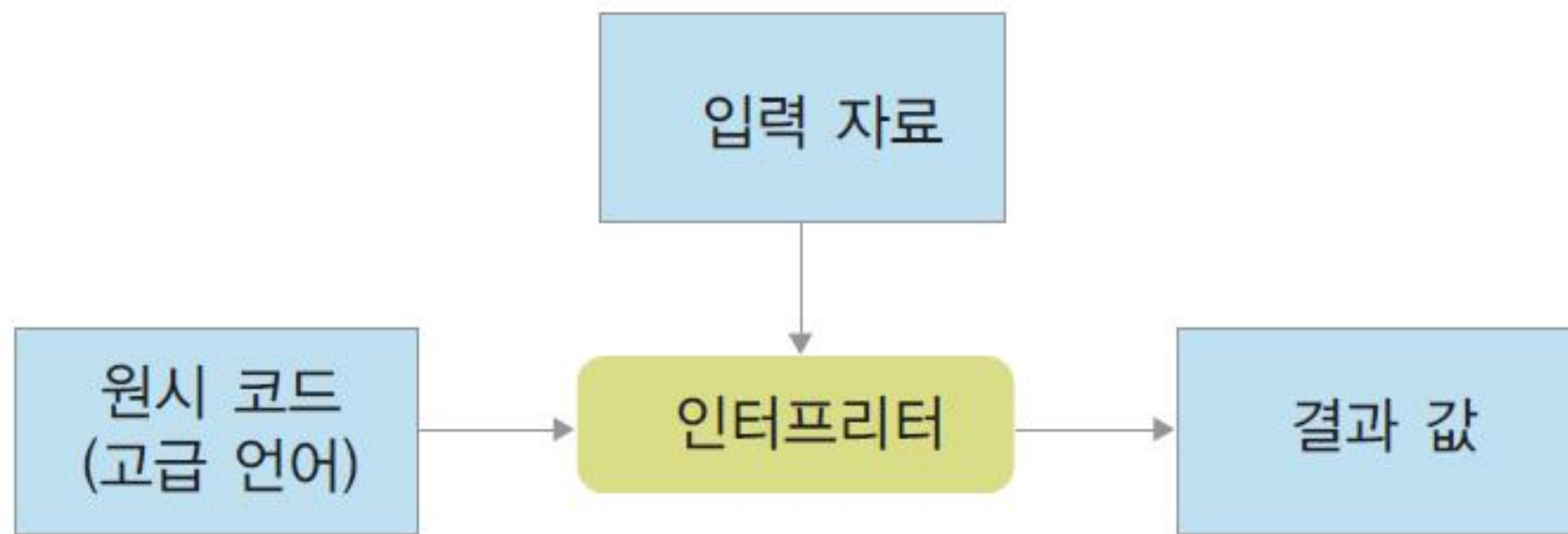


그림 4-9 인터프리터를 이용한 방식

## 2. 프로그래밍 언어의 실행 과정

### 2) 코딩 및 컴파일

#### ◆ 하이브리드 방식

- 컴파일러와 인터프리터를 함께 이용하는 방식
- 리스프, 스노볼4, APL, 프롤로그, 자바 등의 언어에서 사용

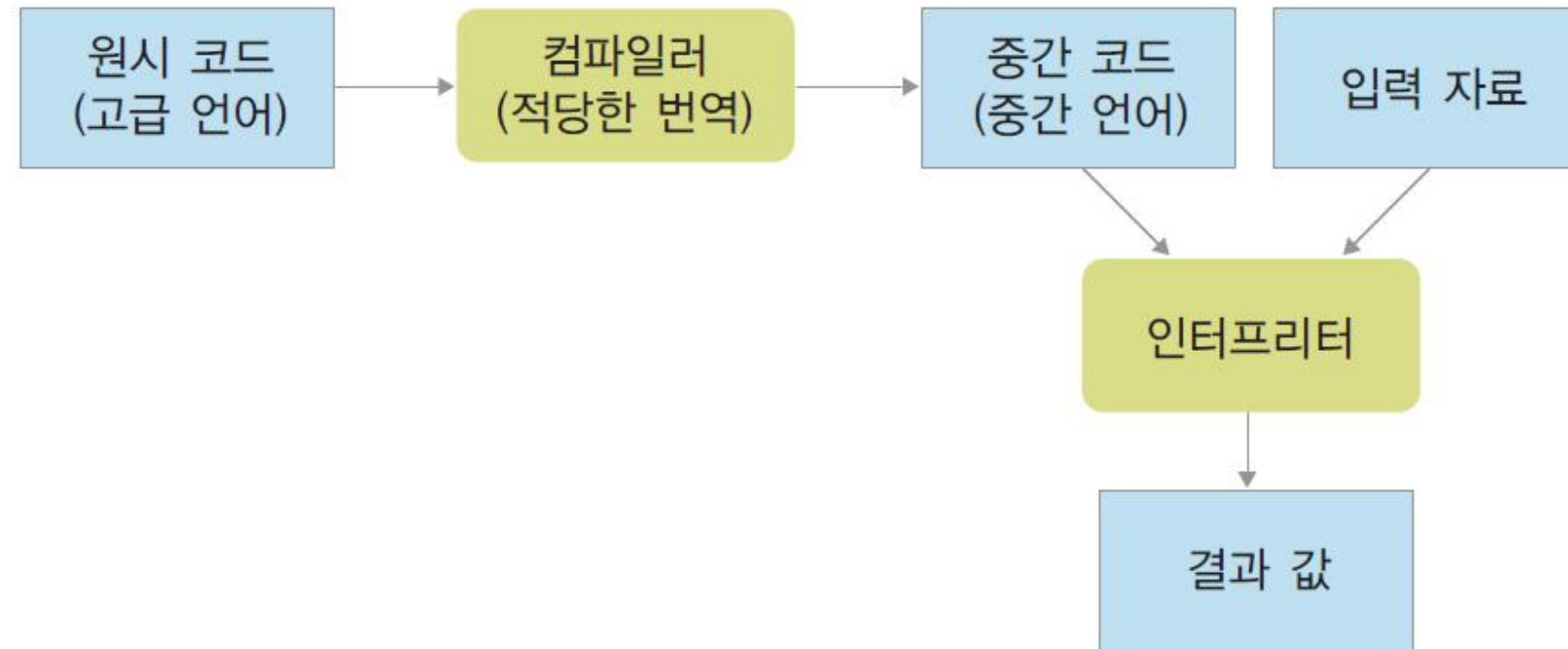


그림 4-10 하이브리드 방식



## 2. 프로그래밍 언어의 실행 과정

### 3) 간단한 프로그램 작성 및 실행

#### ◆ 따라하기 : C 프로그램 작성 및 실행하기

①

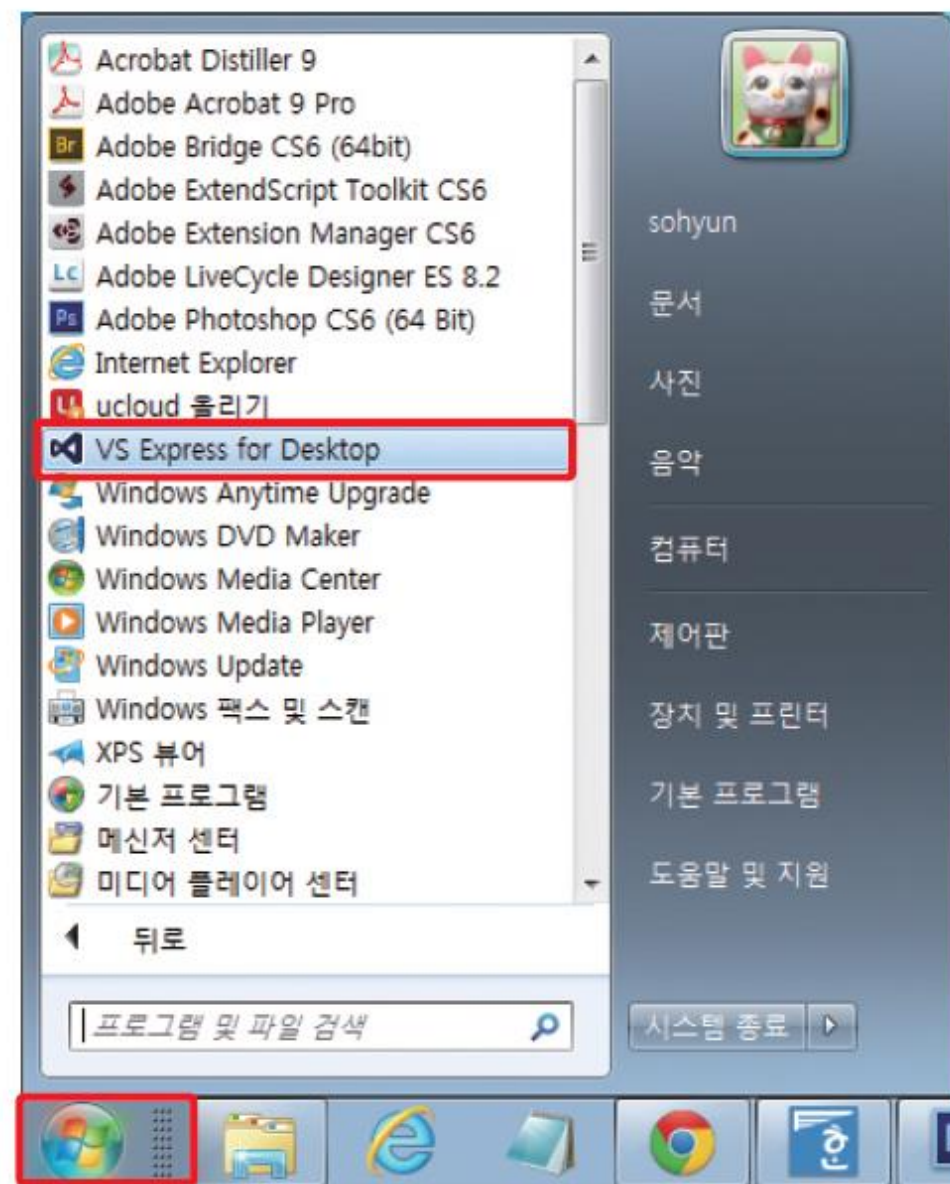


그림 4-11 VS Express for Desktop 실행

## 2. 프로그래밍 언어의 실행 과정

### 3) 간단한 프로그램 작성 및 실행

#### ◆ 따라하기 : C 프로그램 작성 및 실행하기

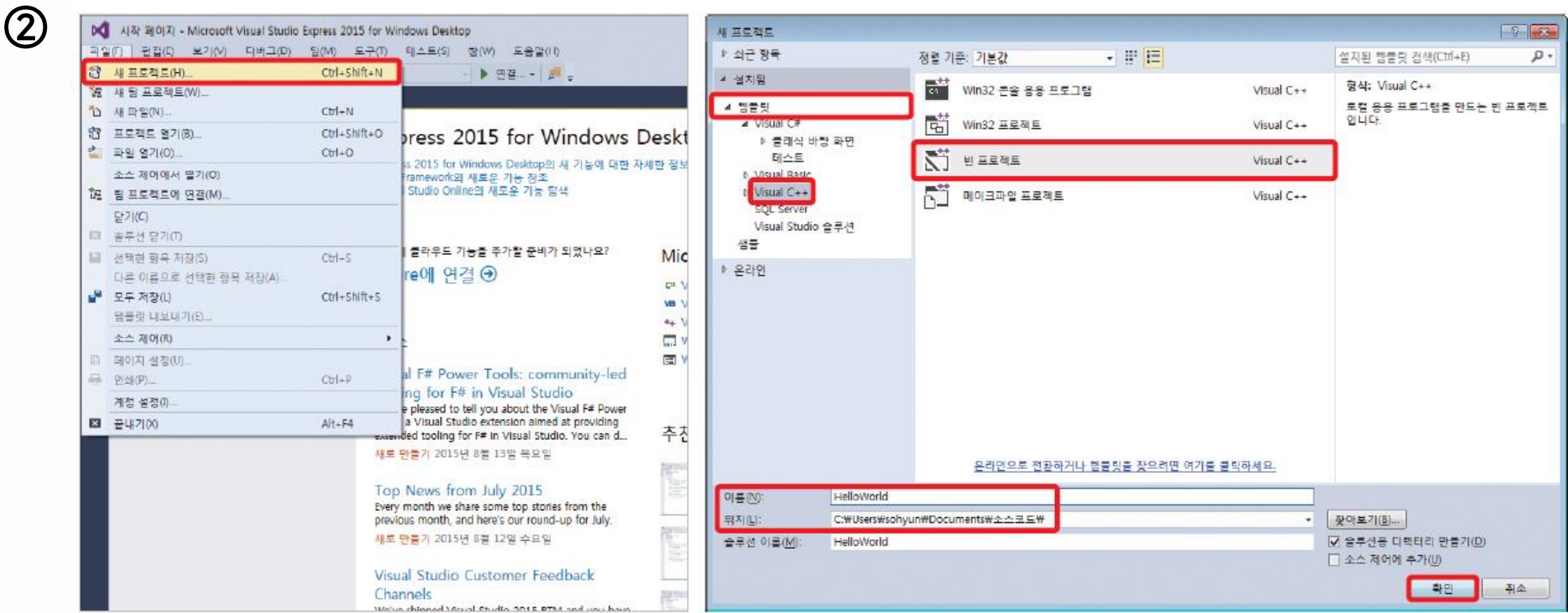


그림 4-12 새 프로젝트 만들기

## 2. 프로그래밍 언어의 실행 과정

### 3) 간단한 프로그램 작성 및 실행

#### ◆ 따라하기 : C 프로그램 작성 및 실행하기

⑤

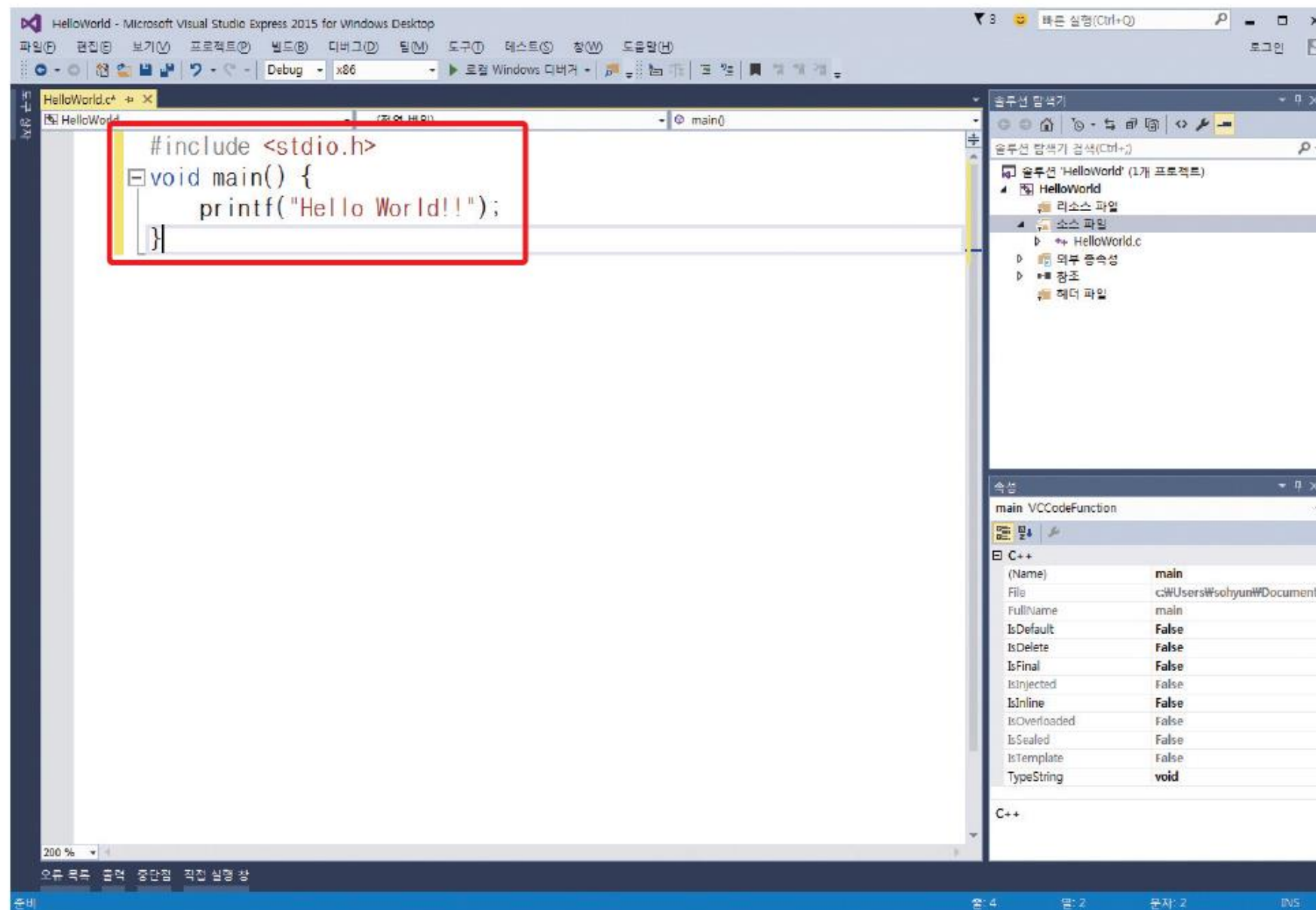


그림 4-15 소스 코드 입력

## 2. 프로그래밍 언어의 실행 과정

### 3) 간단한 프로그램 작성 및 실행

#### ◆ 따라하기 : C 프로그램 작성 및 실행하기

⑥

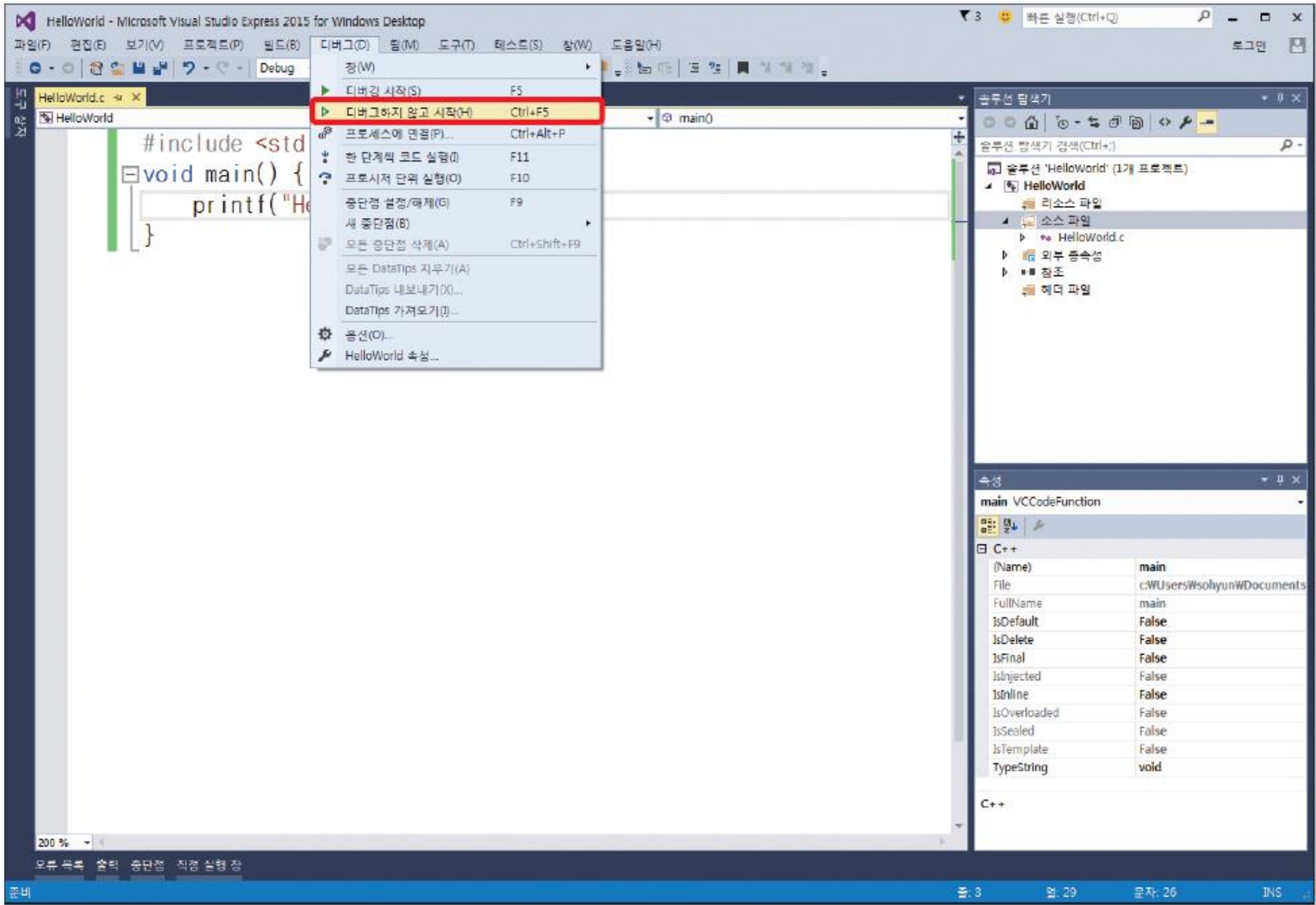


그림 4-16 프로젝트 실행

## 2. 프로그래밍 언어의 실행 과정

### 3) 간단한 프로그램 작성 및 실행

#### ◆ 따라하기 : C 프로그램 작성 및 실행하기

⑦

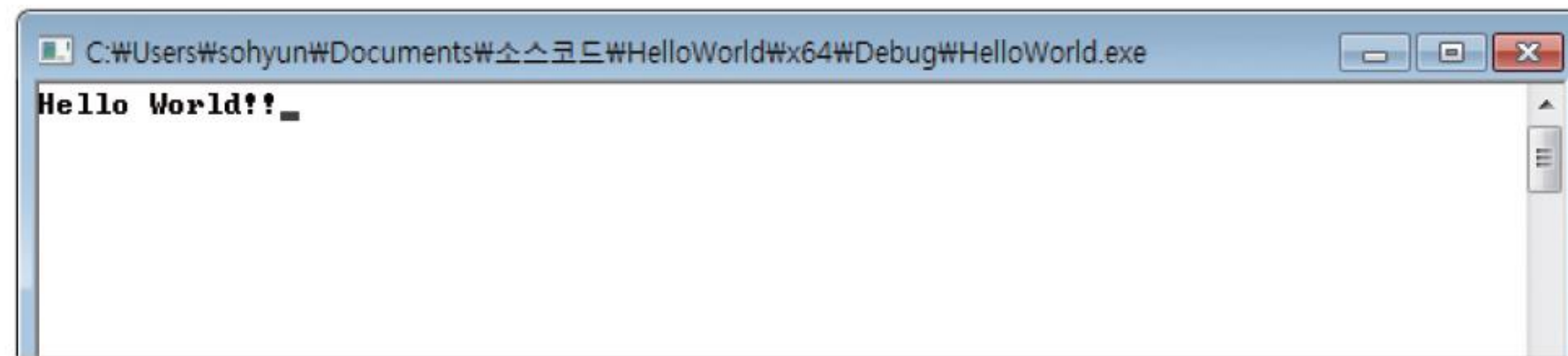


그림 4-17 실행 결과



## 2. 프로그래밍 언어의 실행 과정

### 4) 디버깅 및 시험

#### ◆ 프로그램이 포함하는 모든 오류를 찾아내 제거하는 것

#### ◆ 오류에는 구문 오류와 논리 오류가 있음

- 구문 오류 : 틀린 문자를 입력하거나 문법에 맞지 않는 명령문을 사용했을 때 발생하는 오류
- 논리 오류 : 제어 구조의 부적절한 사용으로 발생하는 오류

#### ◆ 시험은 알파 테스트와 베타 테스트로 구분

- 알파 테스트 : 완성된 프로그램을 개발 환경에서 시험하는 방법
- 베타 테스트 : 특정 고객이 고객에 쓰는 환경에서 시험하는 방법



## 요약

### ◆ 프로그래밍 언어

- 인간과 컴퓨터가 의사소통 할 수 있도록 인간이 컴퓨터에게 내리는 명령, 즉 프로그램을 기술하는 언어이다.

### ◆ 프로그래밍 언어의 유형

- 저급언어: 2진수 형태의 기계어와 기계어를 알기 쉬운 기호로 변형한 어셈블리어가 있다.
- 고급언어: 사람의 표현방식을 이용하여 컴퓨터에게 명령을 내리는 언어로 포트란, 코볼, 파스칼, C언어, C++, 자바 등이 있다.

### ◆ 프로그램 개발과정

- 사용자 요구사항 분석
- 프로그램 설계
- 코딩 및 컴파일
- 디버깅 및 테스트