

본 강의에서 수업자료로 이용되는 저작물은  
저작권법 제25조 수업목적 저작물 이용 보상금제도에 의거,  
한국복제전송저작권협회와 약정을 체결하고 적법하게 이용하고 있습니다.  
약정범위를 초과하는 사용은 저작권법에 저촉될 수 있으므로  
수업자료의 재 복제, 대중 공개·공유 및 수업 목적 외의 사용을 금지합니다.

2024. 8. 30.

부천대학교·한국복제전송저작권협회

C#

C# 개요

# C# 프로그래밍 입문



## 1. 개요

# 1.1 프로그래밍 언어

## ◆ 프로그래밍이란?

프로그래밍이란 컴퓨터를 이용하여 어떠한 계산이나 출력 등의 목적에 맞게 프로그램을 설계하고 만드는 과정을 의미한다. 즉, 필요한 프로그램을 만드는 과정을 프로그래밍이라 한다.

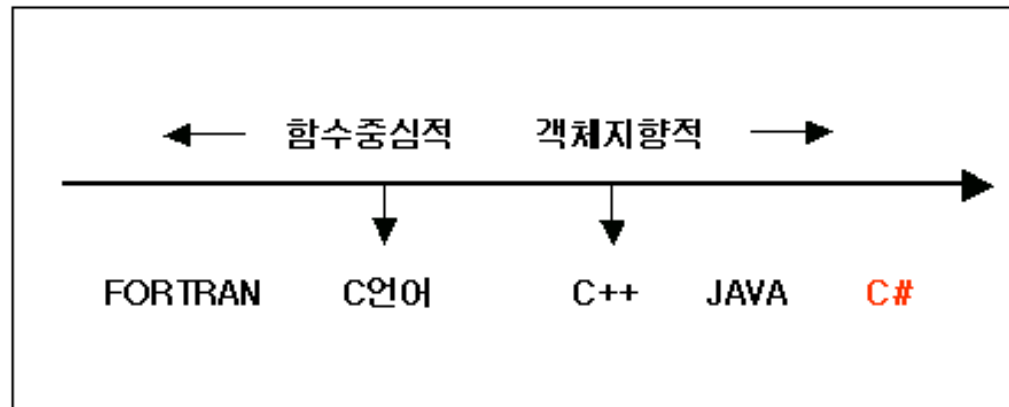
## ◆ 프로그래밍 언어란?

프로그램을 만들고 정상적으로 실행하기 위하여, 프로그램이 실제 실행되는 컴퓨터와 이러한 프로그램을 만드는 프로그래머와의 의사소통 도구를 프로그래밍 언어라 할 수 있다.

# 1.1 프로그래밍 언어

## ◆ 프로그래밍 언어의 역사

오늘날의 C#언어에 이르기까지 사용했던 프로그래밍 언어



# 1.1 프로그래밍 언어

## ◆ 함수중심적 언어와 객체지향적 언어의 비교

함수중심적 언어	객체지향적 언어
BASIC,FORTRAN,PASCAL,C	SMALLTALK,C++,JAVA
함수를 중심으로 이루어져 있음	데이터가 중심으로 이루어져 있음
자료는 인자를 전달받아 처리함	함수는 데이터와 같이 존재할 때 의미 있음

## 1-2. 닷넷 소개

### ◆ 닷넷(.NET)이란?

마이크로소프트에서 2000년 6월 새롭게 발표한 닷넷(.NET)은 지금까지 존재했던 마이크로소프트의 모든 S/W기술을 총망라하고 다른 언어들의 장점을 접목한 새로운 기술

### ◆ 닷넷의 등장 배경

- 소프트웨어 호환성 문제 => 완전한 이식성 제공
  - 소프트웨어도 플랫폼마다 매번 다시 작성해야 했으며 완전한 이식성을 보장 받기 어려기 때문
  - 모든 플랫폼에 두루 실행 할 수 있게 하기 위해서 중간코드 필요
  - 환경 번호가 자바와 닷넷 등장
- 웹 대중화
  - 웹은 분산 환경이므로 서로 다른 하드웨어 환경과 운영체제에서 실행 될 수 있는 플랫폼 독립성이 필수적인 요건

### ◆ 닷넷의 정의

**응용 프로그램이 실행되는 실행 환경이며, 플랫폼이다.**

=> 닷넷은 운영체제가 아니며 운영체제 위에서 돌아가는 실행 환경이다.

## 1-2. 닷넷 소개

### ◆ 새로운 개발 환경

- 소프트웨어는 새로운 것이 발표될 때마다 역호환성이 중요시 되어야 한다.
  - 역호환성 : 아래 버전이 호환되어야 하는 성질
- 닷넷은 과거와의 단절을 선언하고 완전히 새롭게 정의된 개발 방법이다. 하지만 완전히 단절은 아니고 물려 받은 면도 있고 깔끔함을 유지하기 위해 **문법 구조가 명확하고 간결하다.**

### ◆ 닷넷 버전(.NET Framework)의 역사

출시일 년도	개발 툴	닷넷버전(.NET Framework)	C#버전
2002/2/13	VS. NET 2002	닷넷1.0	C#1.0
2003/4/24	VS .NET 2003	닷넷1.1	C#1.2
2005/11/7	VS 2005	닷넷2.0	C#2.0
2007/11/19	VS 2008	닷넷3.5	C#3.0
2010/4/12	VS 2010	닷넷4.0	C#4.0
2013/10/17	VS2013	닷넷4.5	C#5.0
2015/7/20	VS2015	닷넷4.6	C#6.0
2017/3/7	VS2017	닷넷4.7	C#7.0
2019/4/2	VS2019	닷넷4.8 ~ 닷넷5.0	C#8.0~C#9.0
2021/11/8	VS2022	닷넷6.0	C#10.0

▪프레임워크(Framework) : 대규모 라이브러리로 프로그램의 기본 틀을 모두 제공

▪플랫폼(Platform): 응용 프로그램들을 실행하는데 사용되는 하드웨어와 소프트웨어의 집합

▪닷넷 프레임워크(.NET Framework) : 닷넷 플랫폼(실행 엔진)+ 클래스 라이브러리

▪닷넷을 사용할 수 있는 환경을 제공(윈도우 프로그램을 개발하고 실행 할 수 있도록 서비스 집합과 실행 환경 제공: 윈도우 환경 전용)



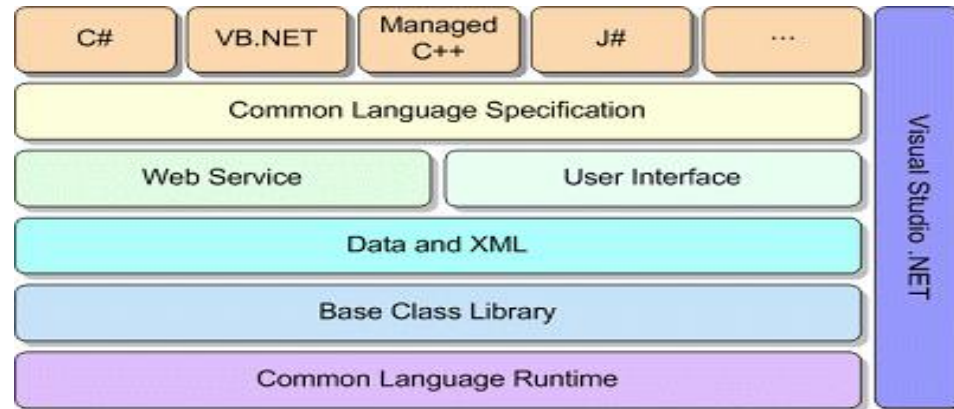
## 1-3. 닷넷의 구조

### ◆ 닷넷(.NET)의 특성

- 플랫폼의 독립성 : 한 번 작성된 프로그램은 운영체제에 상관없이 실행 가능
  - IL이라는 중간코드 생성
- 언어의 독립성 : 닷넷 응용프로그램은 어떤 언어로나 개발 가능하고, 여러 언어를 섞어서 하나의 프로그램을 개발하는 것이 가능하며 언어끼리 자유롭게 서로의 함수를 호출도 가능

# 1-3. 닷넷의 구조

## ◆ 닷넷의 구성 요소



- Visual Studio .NET : C# 등의 닷넷 언어를 만들기 위한 도구로 통합 개발 환경
  - 코드 작성, 컴파일 및 실행 등과 같은 작업과 참조, 데이터 연결, 폴더 및 파일과 같이 개발 작업에 필요한 관련 항목들을 모두 관리 => **통합 개발 환경**
- 닷넷 프레임워크 : 닷넷을 사용할 수 있는 환경을 제공(서비스 집합과 프로그램 실행 환경: 윈도우 환경 전용)
  - **CLR(Common Language Runtime)** :닷넷 실행 엔진, 닷넷 플랫폼
  - BCL(Base Class Library) :잘 정리된 클래스의 거대한 집합
  - CTS(Common Type System) :모든 닷넷 언어들의 공통으로 사용하는 타입 체계를 정의
  - CLS(Common Language Specification) :닷넷용 언어가 되기 위한 최소한의 규약
    - CLS를 만족하면 서로 다른 언어에서 만들어진 라이브러리를 공유할 수 있다.

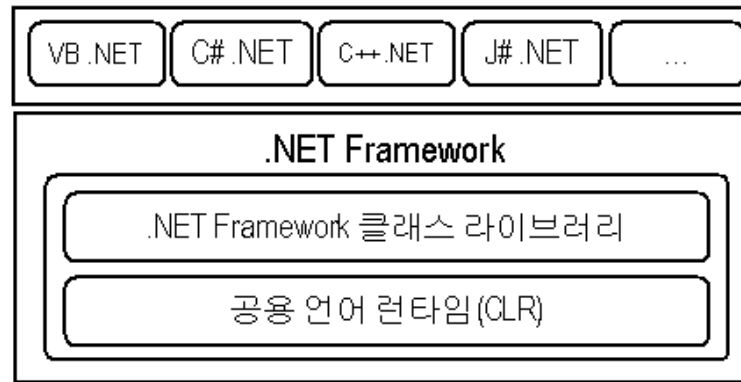
## 1-3. 닷넷의 구조

### ◆ 닷넷(.NET)의 장점

- 플랫폼의 독립성
- 언어 상호 운용성 : 언어 독립성보다 더 상위 개념으로 다른 기종의 언어끼리 교차 호출, 교차 상속, 교차 디버깅 가능 => 자바 불가능
- 완전한 객체 지향 언어: 생산성이 높다
- 응용 프로그램 배포하기 쉽다
- 보안 적용이 쉽고 완벽하다
- 프로그래밍 환경이 쾌적=> 가비지 컬렉션 기능 제공
- 닷넷의 활용 범위가 광범위하다
- 방대한 양의 클래스 라이브러리(BCL)제공

## 1-4. 닷넷 프레임워크 구성 요소

### ◆ 닷넷 프레임워크의 구성 요소

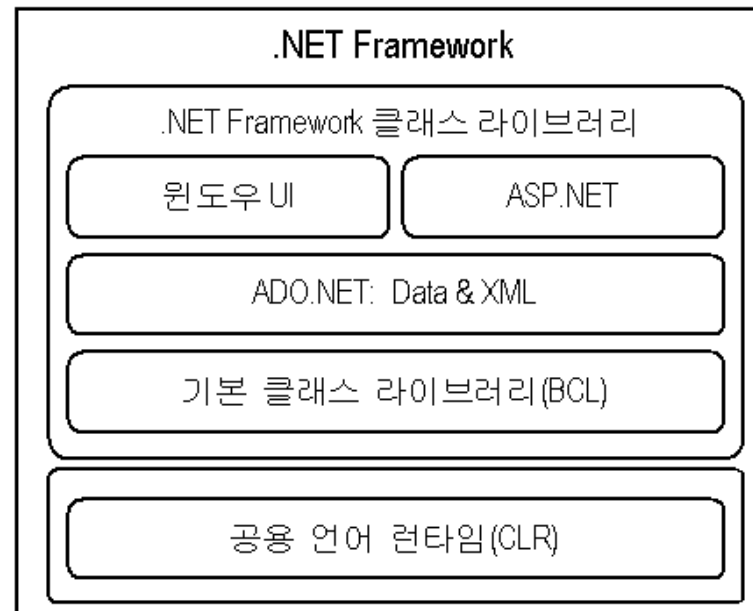


**닷넷 프레임워크**는 크게 두 가지로 나눌 수 있는데, 프로그램을 개발을 할 때 필요한 서비스들의 집합과 프로그램 실행환경으로 구성된다.

- **.NET Framework 클래스 라이브러리**
- **CLR(Common Language Runtime) : 닷넷 실행 엔진, 닷넷 플랫폼**

## 1-4. 닷넷 프레임워크의 구성 요소

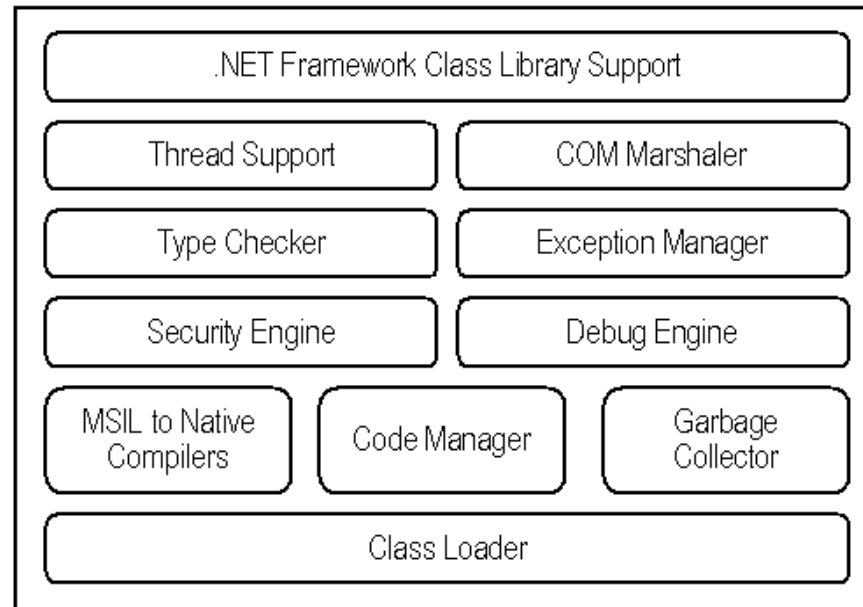
### ◆ .NET Framework 클래스 라이브러리



- BCL : 클래스 라이브러리 중 가장 핵심적인 기능을 하는 클래스의 모음
- ADO.NET : 데이터베이스를 사용할 때 지원하는 클래스
- 윈도우 UI : 닷넷 언어로 윈도우 프로그램을 만들 때 지원하는 클래스
- ASP.NET : 닷넷 언어로 웹 프로그래밍을 할 때 지원하는 클래스

## 1-4. 닷넷 프레임워크의 구성 요소

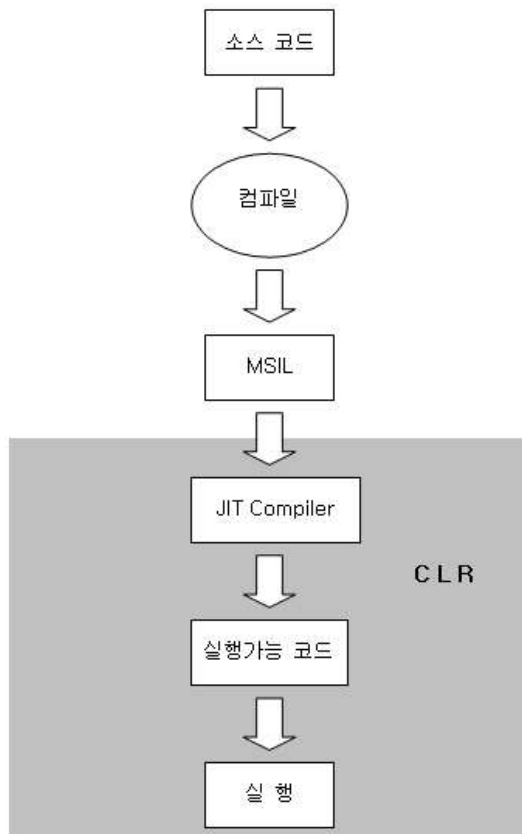
### ◆ CLR(Common Language Runtime)



- CLR의 주요 기능은 JIT(Just-in-Time) 컴파일러를 통해 MSIL 코드(중간코드)를 네이티브 코드(실행가능 코드, 기계어)로 변환한다
- JIT 컴파일러 : 플랫폼 독립성 => 중간 코드를 기계어로 컴파일, 자바의 가상머신(JVM)과 동일
- MSIL : 중간 코드 => 자바의 바이트 코드와 동일

## 1-5. CLR의 역할

### ◆ CLR의 역할

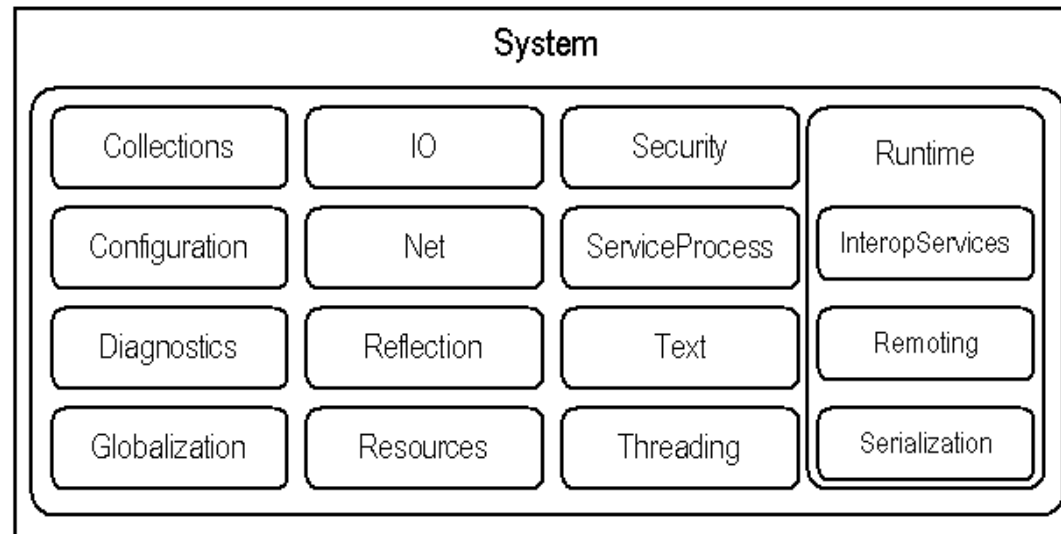


CLR은 닷넷언어로 만들어진 코드를 읽고 해석하여 실행하는 일종의 가상머신(Virtual Machine)이다. Visual Basic .NET, C# 등과 같은 닷넷 언어는 윈도우 운영체제의 네이티브 코드(native code)로 컴파일되지 않고, 마이크로소프트 중간 언어(MSIL, MicroSoft Intermediate Language)로 컴파일된다.

따라서 닷넷 애플리케이션은 윈도우에서 바로 실행할 수 없는데, 닷넷 프레임워크의 CLR에 있는 JIT(Just In Time) 컴파일러가 MSIL 코드를 네이티브 코드로 변환해 실행 가능하게 해준다.

# 1-6. .NET Framework 클래스 라이브러리

## ◆ BCL(Base Class Library)

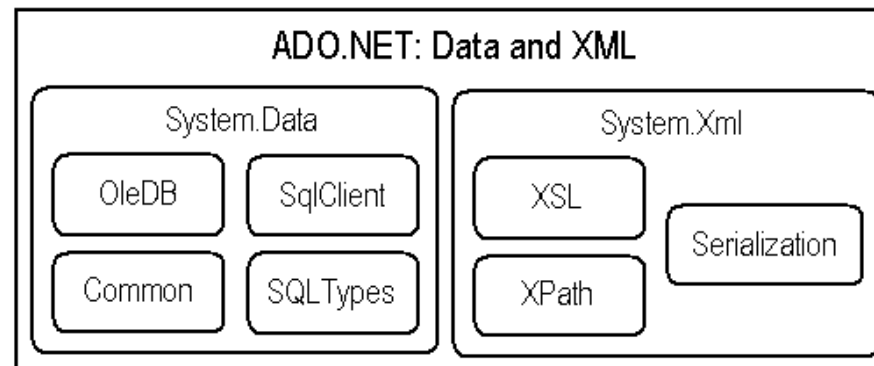


BCL은 기존 마이크로소프트의 다양한 개발 라이브러리를 닷넷에서 사용할 수 있게 통합해 놓은, 닷넷 언어를 위한 기본적인 클래스 라이브러리의 모음을 말한다



# 1-6. .NET Framework 클래스 라이브러리

## ◆ ADO.NET & XML



- ADO.NET은 닷넷 프레임워크의 일부로 System.Data 네임스페이스와 하위 네임스페이스에서 제공하는 데이터 접근 기술로 이루어짐
- System.Data 네임스페이스는 ADO.NET 객체 모델을 구성하는 클래스로 이루어져 있다.
- System.Xml 네임스페이스는 W3C 호환 XML 파서와 XML 작성기 (XMLWriter), 그리고 XSLT와 XPath 등 많은 XML 관련 기술을 위한 클래스로 이루어져 있다.

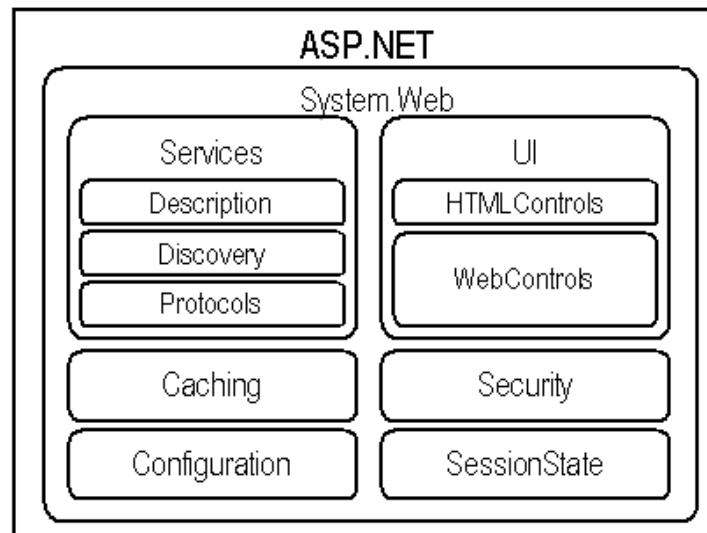
# 1-6. .NET Framework 클래스 라이브러리

## ◆ 윈도우 UI

윈도우 폼 생성에 관련된 클래스는 `System.Windows.Forms` 네임스페이스에 포함되어 있으며, 윈도우 폼 유형은 크게 표준 윈도우, MDI 윈도우, 대화상자로 나눌 수 있다

# 1-6. .NET Framework 클래스 라이브러리

## ◆ ASP.NET



- ASP.NET은 강력한 웹 애플리케이션을 개발하기 위한 프로그래밍 프레임워크
- ASP.NET 웹 폼(Web Forms)은 쉬우면서도 강력한 방법으로 웹 사용자 인터페이스 (Web UI) 페이지를 생성할 수 있도록 한다

# 1-7. c# 소개

## ◆ C# 프로그래밍 언어

- MS사의 앤더스 헬스버그(Anders Hejlsberg)가 고안
- .NET에 최적화된 언어
- 컴포넌트 지향 프로그래밍 언어
- 자바의 단점을 보완
  - 실행 방법: 자바: 인터프리테이션, C#: 컴파일 방법
  - 자바 언어를 대체할 수 있는 언어

## ◆ C#의 특징

- 객체지향 언어: 자료 추상화
- 델리게이트와 이벤트
- 멀티스레드, 예외처리
- 연산자 중복, 제네릭

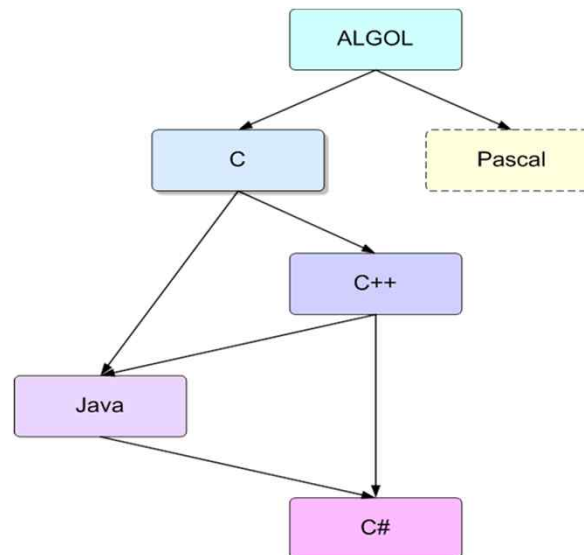
# 1-7. c# 소개

## ◆ C 계열의 언어

➤ C++와 자바로부터 영향을 받았음.

- C: 연산자와 문장 등 기본적인 언어의 기능
- C++: 객체지향 속성, 연산자 중복, 제네릭(Generic)
- 자바: 예외처리와 스레드

## ◆ C# 언어의 계통도



# 1-7. c# 소개

## ◆ C# 이란?

C#은 기존 언어 중에 가장 막강한 기능을 자랑하는 C++을 기반으로, JAVA의 장점을 혼합하고, 닷넷이라는 새로운 개념을 도입하여 만들어진 언어

C#은 C++처럼 강력하고, JAVA처럼 유연하다.

## 1.8 C# 프로그램 시작하기

### ◆ C# 개발 환경

- SDK를 이용 – 편집기, 컴파일러, 실행엔진, 클래스 라이브러리
- 통합개발 환경(IDE: Integrated Development Environment)
  - **Visual Studio 2022**

### ◆ 콘솔 응용 프로그램

- 문자기반 명령어 프롬프트 환경에서 실행
- 키보드를 통해 입력, 화면에 문자로 출력
- 개발환경
  - 방법1: 메모장 이용
  - **방법2: Visual Studio 2022**

### ◆ 윈폼 응용 프로그램

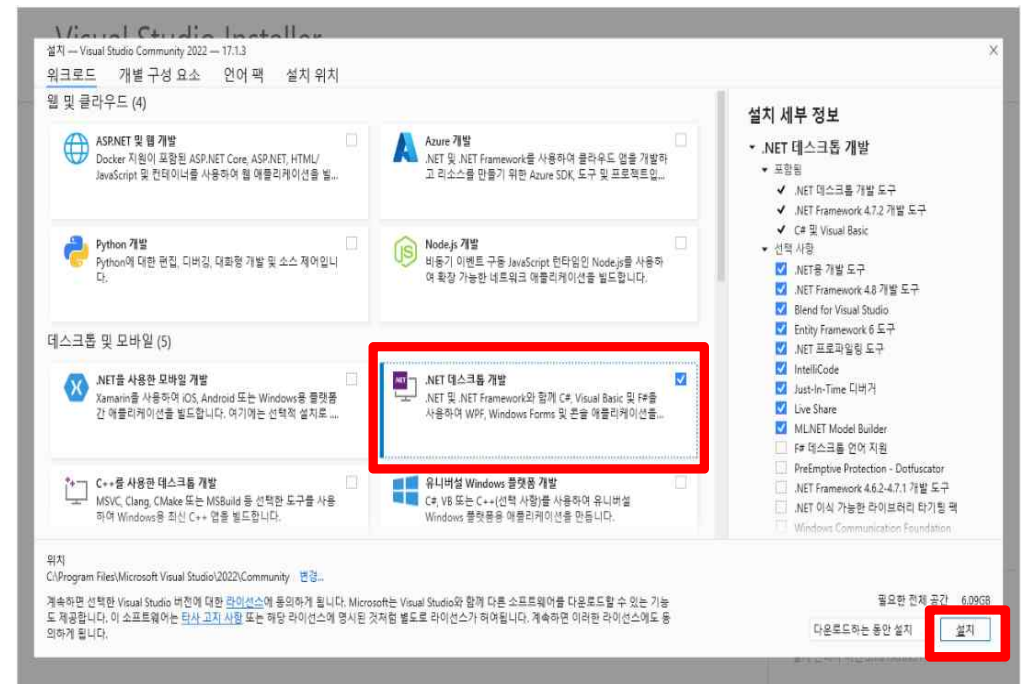
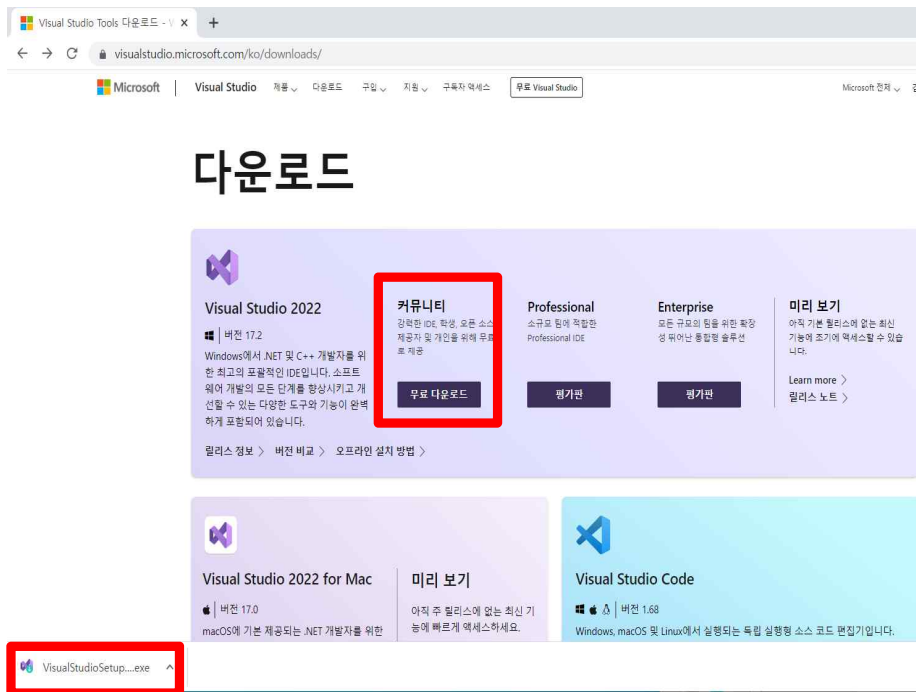
- 윈도우 폼 애플리케이션(Windows forms Application)의 약어
- GUI를 통해서 입출력을 수행
- 이벤트 처리 방식을 통해 실행
- 개발 환경
  - **방법1: Visual Studio 2022**

## 1.8 C# 프로그램 시작하기(VS 2022)

### ◆ Visual Studio 2022 –C# 설치 (처음 VS 2022 설치 할 때)

▶ <https://visualstudio.microsoft.com/ko/downloads/> 에서 ‘Visual Studio 2022 커뮤니티 버전’ 무료다운로드 버튼을 클릭하여 설치 파일을 다운 받고 실행

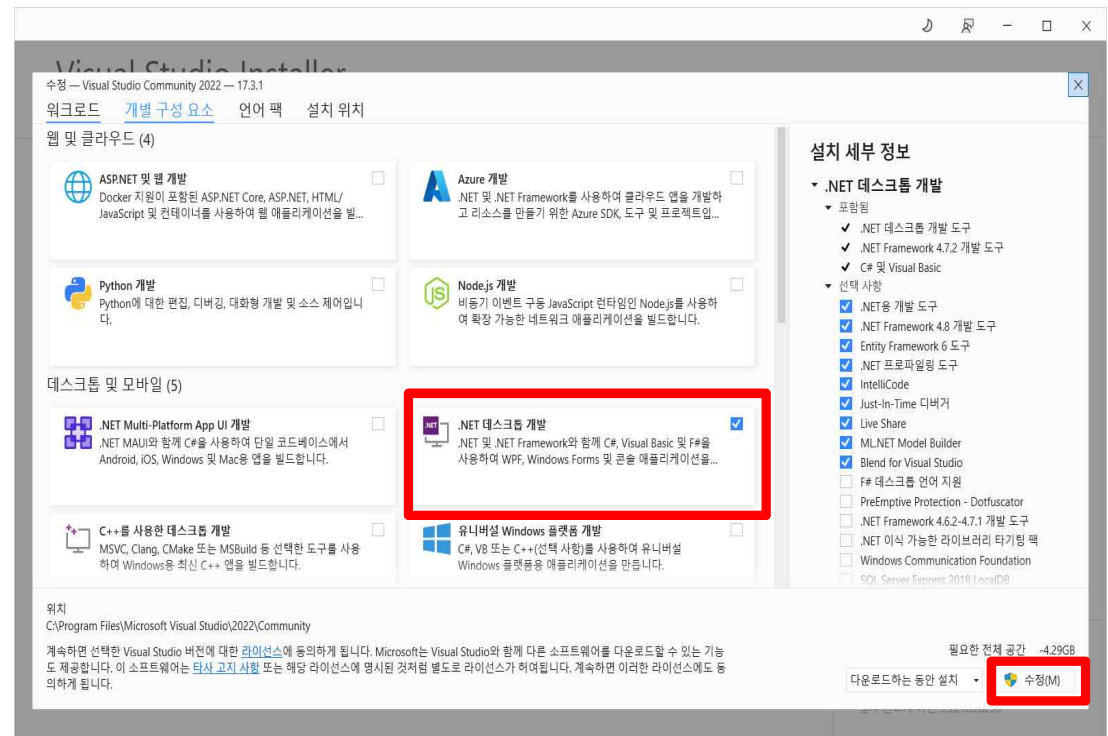
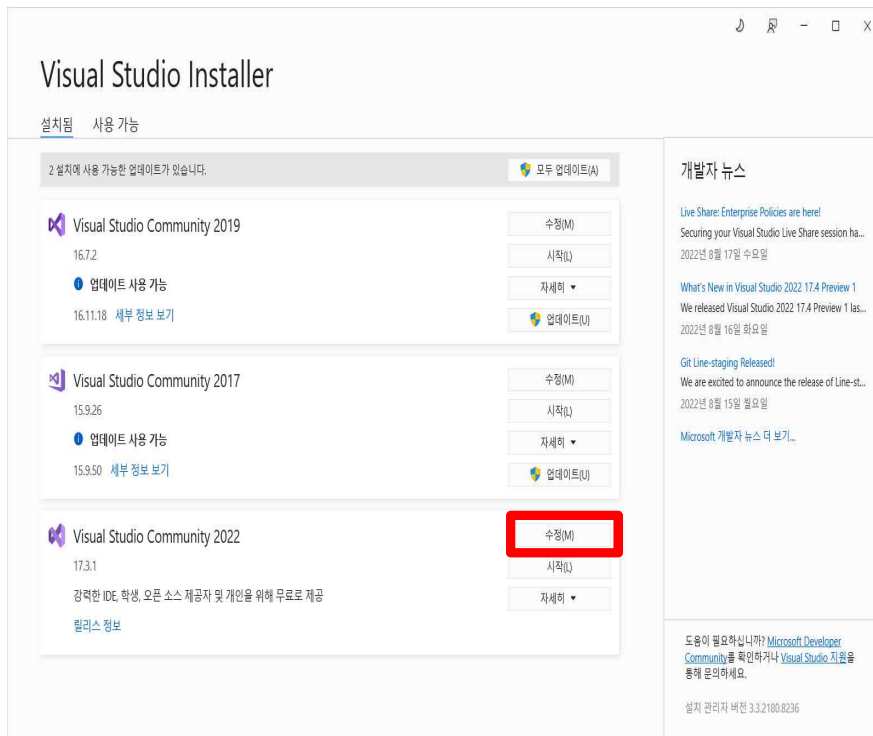
▶ 설치할 때 다음과 같은 팩 설치 화면이 나오면 [데스크톱 및 모바일]-[.NET 데스크톱 개발]를 체크하고 설치 버튼 클릭





## 1.8 C# 프로그램 시작하기

- ◆ Visual Studio 2022 –C# 설치 (다른 언어 사용으로 VS 2022 설치가 되어 있을 때)
  - ▶ [Visual Studio Installer(설치 관리자)]에서 VS 2022에서 수정 버튼을 클릭
  - ▶ 다음과 같은 팩 설치 화면이 나오면 [데스크톱 및 모바일] - [.NET 데스크톱 개발]를 체크하고 설치



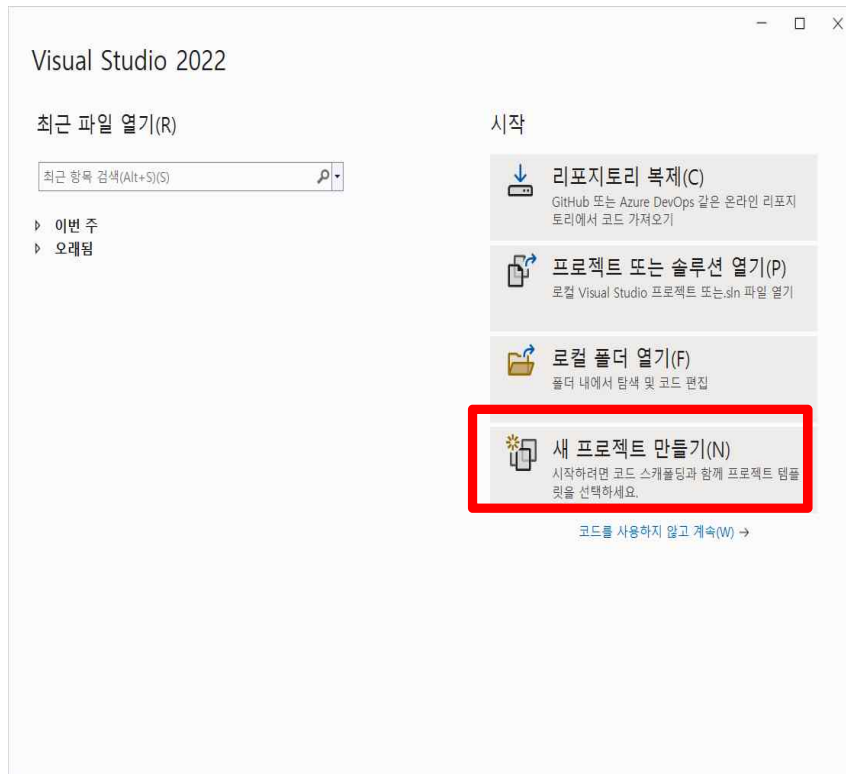
# 1.8 C# 프로그램 시작하기

## ◆ Visual Studio 2022 –C# 실행

▶ **마이크로소프트 계정이 없다면 먼저 회원 가입한 후 바로 로그인**

▶ **C# 프로그램을 개발하기 위해 [새 프로젝트 만들기]선택한 후**

**C# 콘솔 프로그램(콘솔 앱(.NET Framework))이나 C# 윈폼 프로그램(Windows Form 앱(.NET Framework)) 작성**



## 1.8 C# 프로그램 시작하기

### ◆ 콘솔 프로그램 작성하기 맛보기 –Visual Studio 2022

1. 시작 - 모든 프로그램 - visual studio 2022

2. [새프로젝트 만들기] 창에서 다음을 설정 후 확인

-언어 종류 설정: **c#**

-프로젝트 종류 선택 : 콘솔 앱(.NET Framework)

-이름 : w1\_ex1\_c\_xxxx(학번 앞자리 두자리와 학번 끝자리 두자리)(예:1900)

-위치 : c:\hong

-솔루션이름:w1\_xxxx(학번 앞자리 두자리와 학번 끝자리 두자리)(예:1900)

3. Program.cs에서 Main()메서드에 다음을 입력

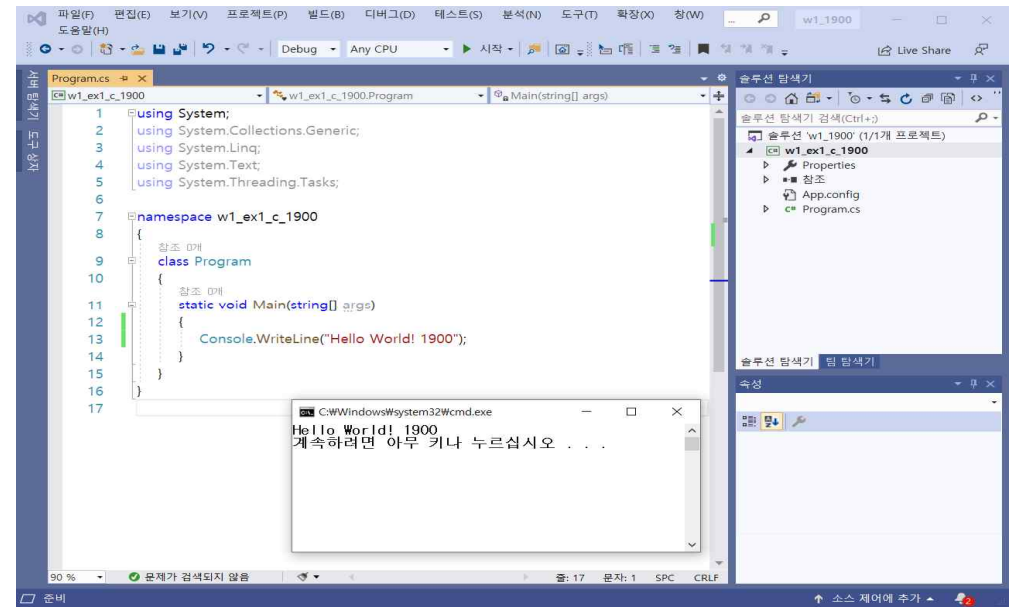
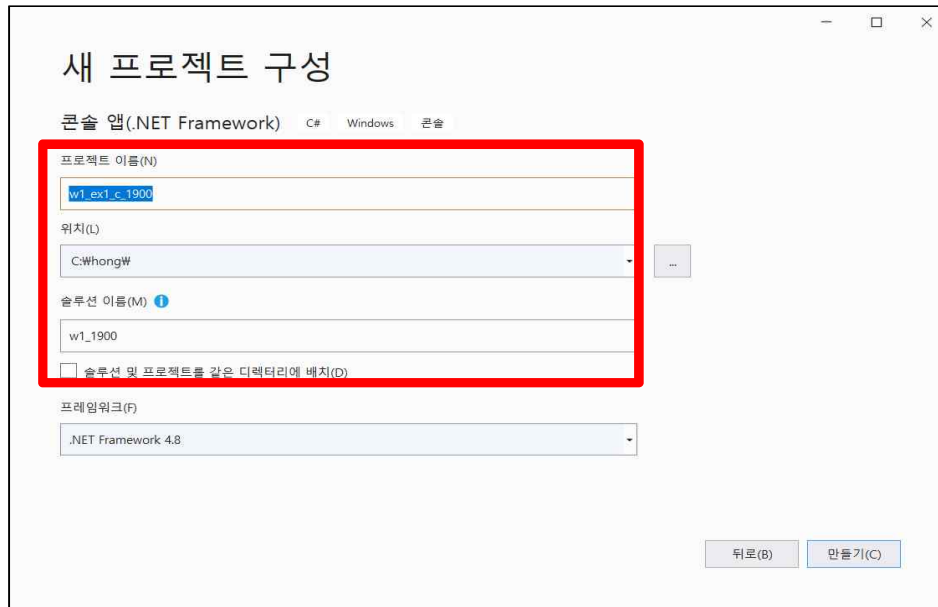
**Console.WriteLine("Hello World! xxxx");** 과 같은 처리 문장을 입력

4. 컴파일; 빌드 –솔루션 빌드

5. 실행: 디버그 –디버그 하지 않고 시작 또는 ctrl + F5를 누른다.

## 1.8 C# 프로그램 시작하기

### ◆ 콘솔 프로그램 작성하기 맛보기 – Visual Studio 2022



- ◆ Visual Studio 2022 색 테마는 [파랑]으로 설정
  - ▶ 도구 => 옵션 => 환경 => 일반 => 색 테마 => 파랑
- ◆ Visual Studio 2022 텍스트편집기는 글꼴을 [굵게]로 설정
  - ▶ 도구 => 옵션 => 환경 => 글꼴 및 색 => 굵게
- ◆ 명령프롬프트 화면 배경은 [흰색]으로 설정
  - ▶ 제목 표시줄(마우스오른쪽 버튼 클릭) => 속성 => 색 => 화면 배경과 화면 텍스트 변경

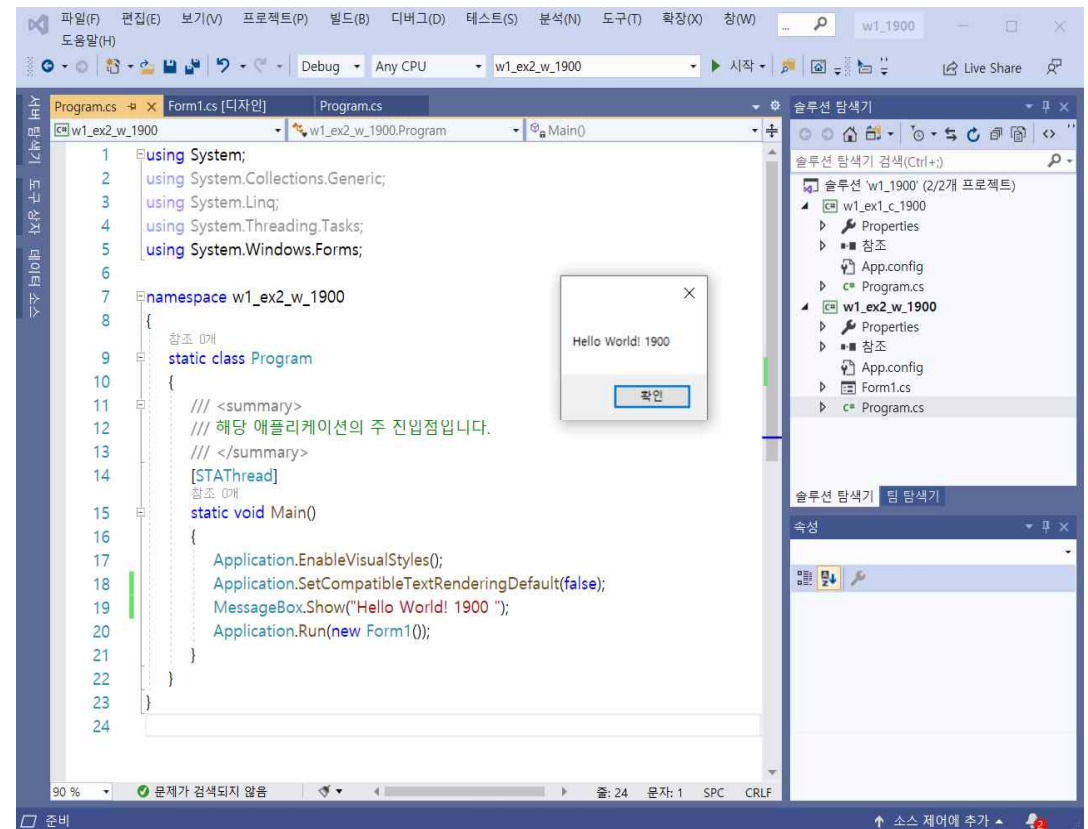
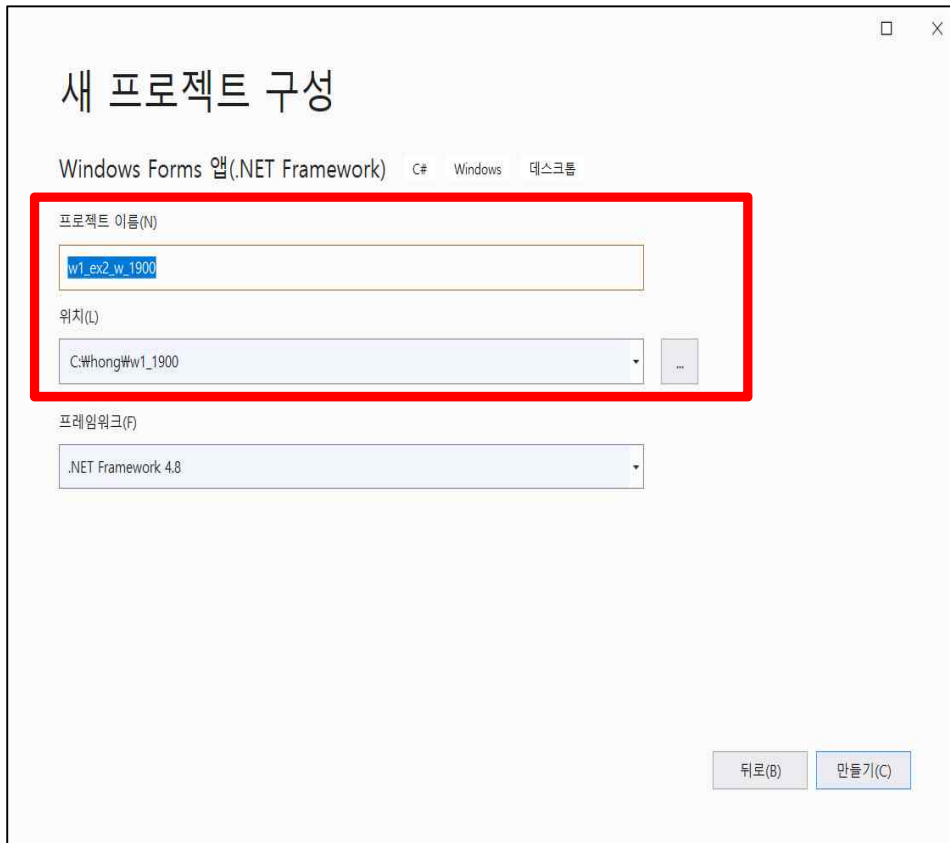
## 1.8 C# 프로그램 시작하기

### ◆ windows 응용프로그램 맛보기 -Visual Studio 2022

6. 같은 솔루션에 새로운 프로젝트를 추가하는 방법으로 windows 응용프로그램 만들기  
메뉴에서 [파일]-[추가]-[새프로젝트] 클릭  
[새프로젝트 추가] 창에서 다음을 설정 후 확인
  - 언어 종류 설정: **c#**
  - 프로젝트 종류 선택 : **Windows Forms 앱(.NET Framework)**
    - 이름 : **w1\_ex2\_w\_xxxx**(학번 앞자리 두자리와 학번 끝자리 두자리)(예:1900)
    - 위치 : **c:\hong\w1\_xxxx**(학번 앞자리 두자리와 학번 끝자리 두자리)(예:1900)
7. Program.cs에서 Main()메서드 안에서  
“Application.Run(new Form1());” 문장 앞에 다음 문장을 입력  
**MessageBox.Show( “Hello World! xxxx ”);**
8. 솔루션탐색기창에서 실행하고자 하는 프로젝트를 선택한 후 마우스 오른쪽 버튼을 눌러  
단축메뉴에서 **“시작 프로젝트로 설정”** 을 선택한다
9. 컴파일; 빌드 -솔루션 빌드
10. 실행: 디버그 -디버그 하지 않고 시작 또는 ctrl + F5를 누른다.

## 1.8 C# 프로그램 시작하기

### ◆ windows 응용프로그램 맛보기 –Visual Studio 2022



## 1.9 C# 프로그램 실행과정

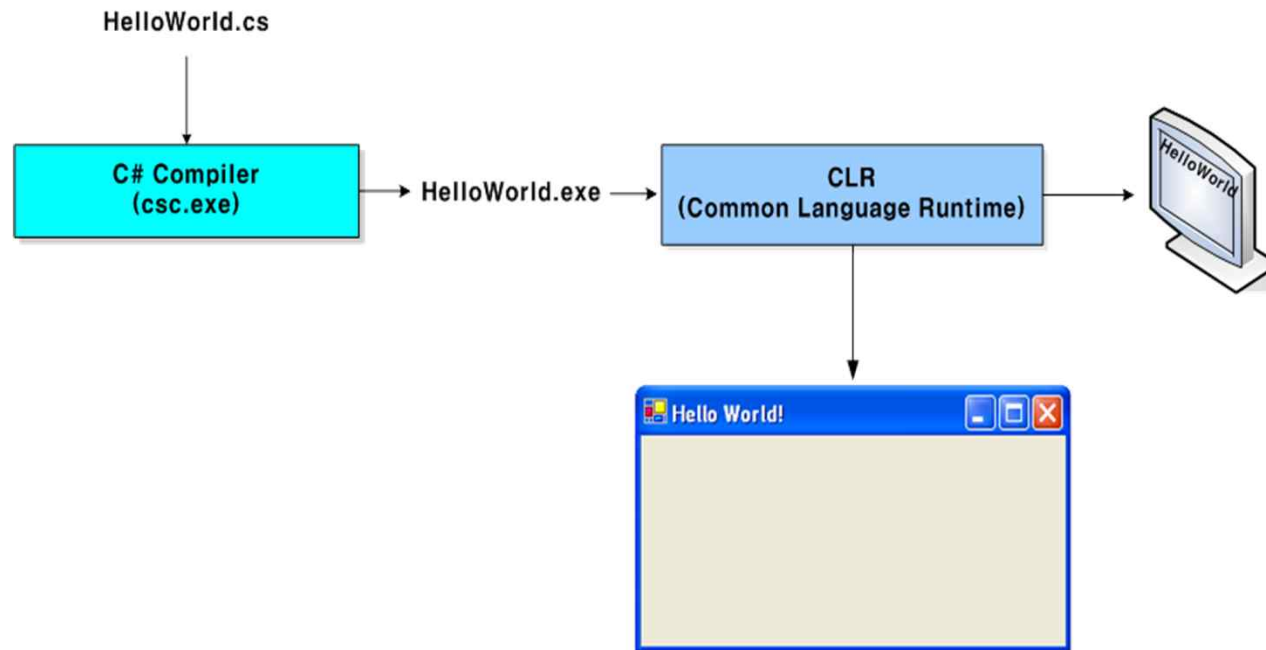
### ◆ 컴파일 과정

- ▶ csc : C# compiler

### ◆ 실행 시스템

- ▶ CLR – Common Language Runtime

### ◆ 실행 과정



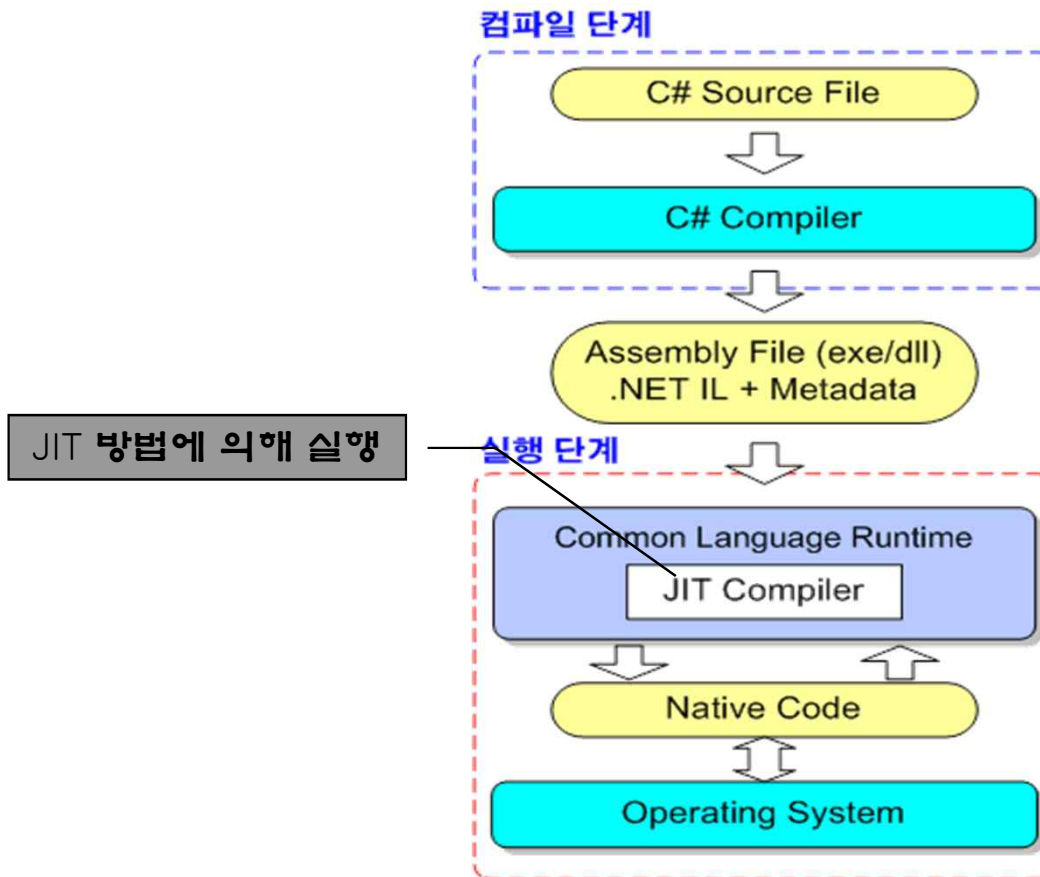
## 1.9 C# 프로그램 실행과정

### ■ 컴파일 단계

■ C# Compiler

### ■ 실행 단계

■ Common Language Runtime





# 1.10 C# 프로그램 구조

## ◆ C# 프로그램 구조

