



프로그래밍 활용

Chapter 2

C# 프로그램 기초

오 세 종

 DANKOOK UNIVERSITY

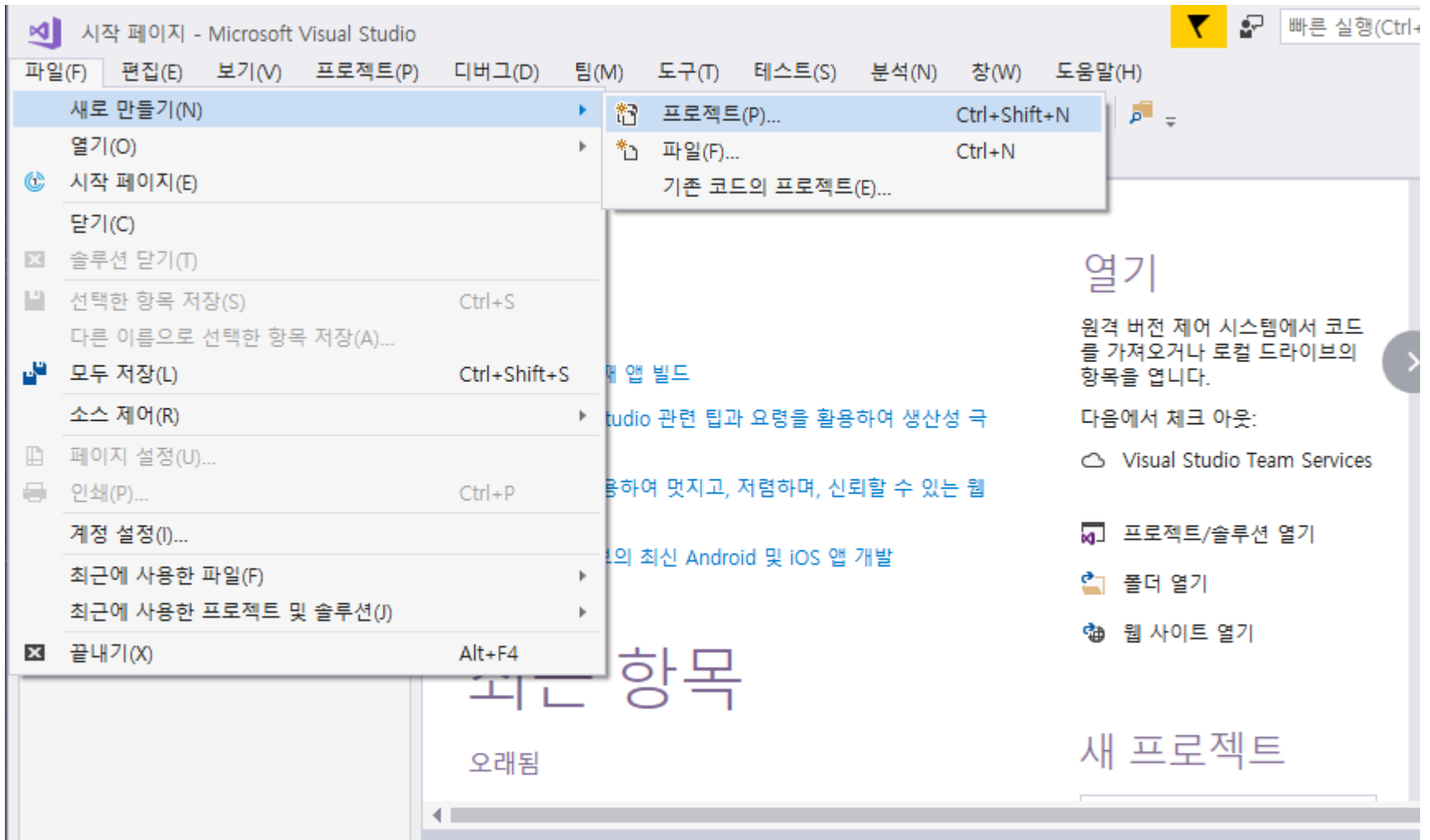
Contents



1. Hello world
2. C# 프로그램 기초

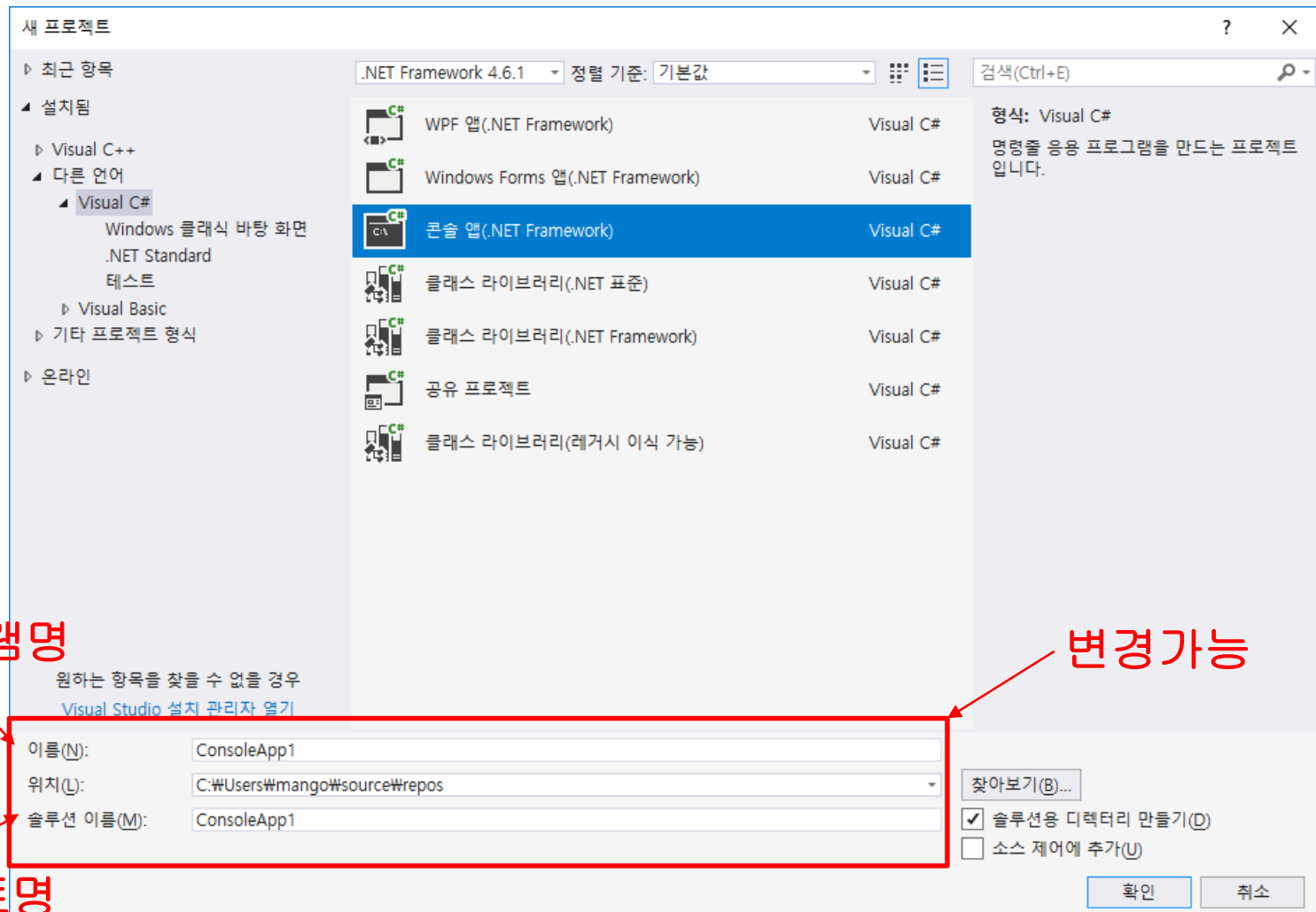
1. Hello world

- Visual studio 실행 : 프로젝트 만들기



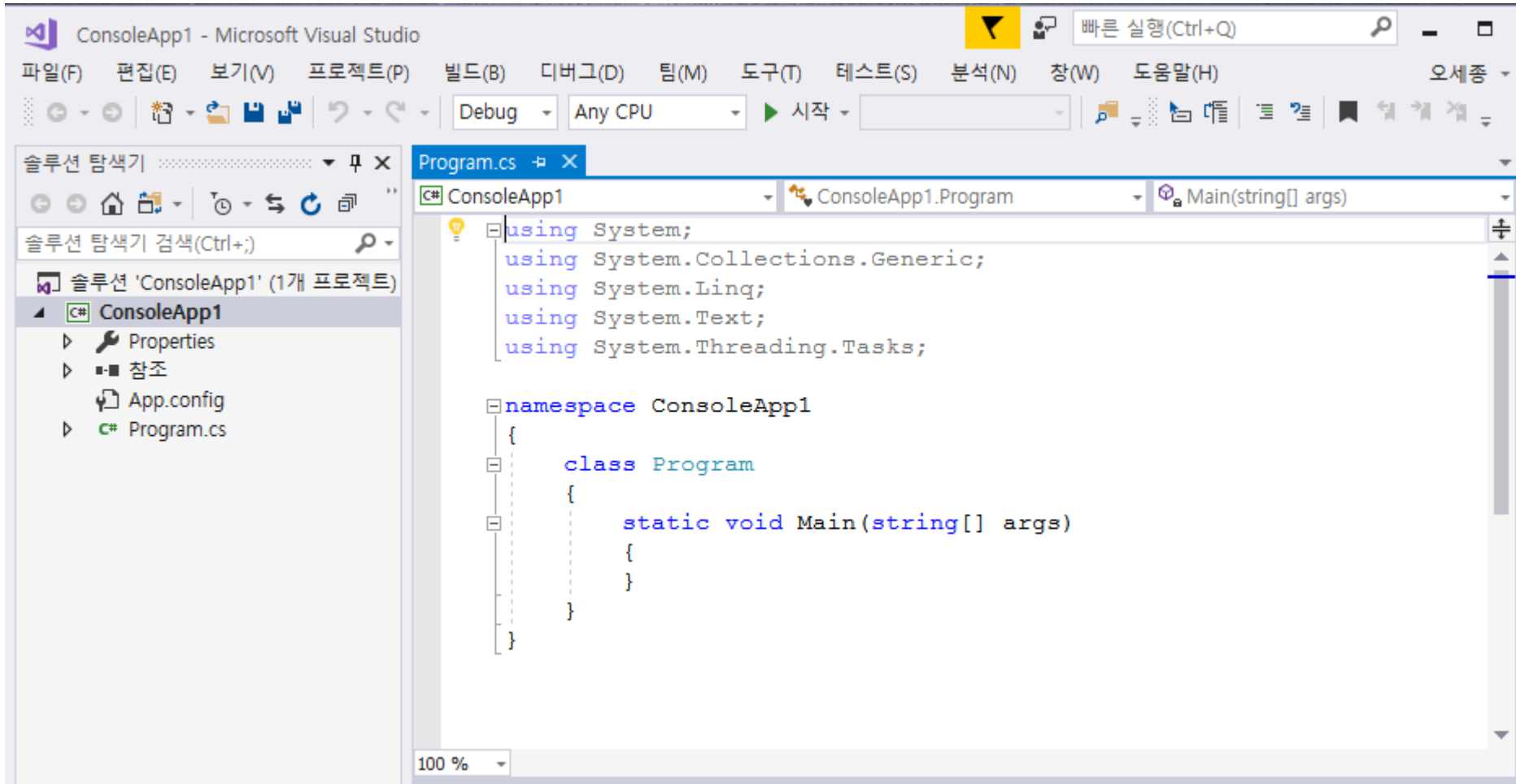
1. Hello world

- C# -> 콘솔 앱 선택



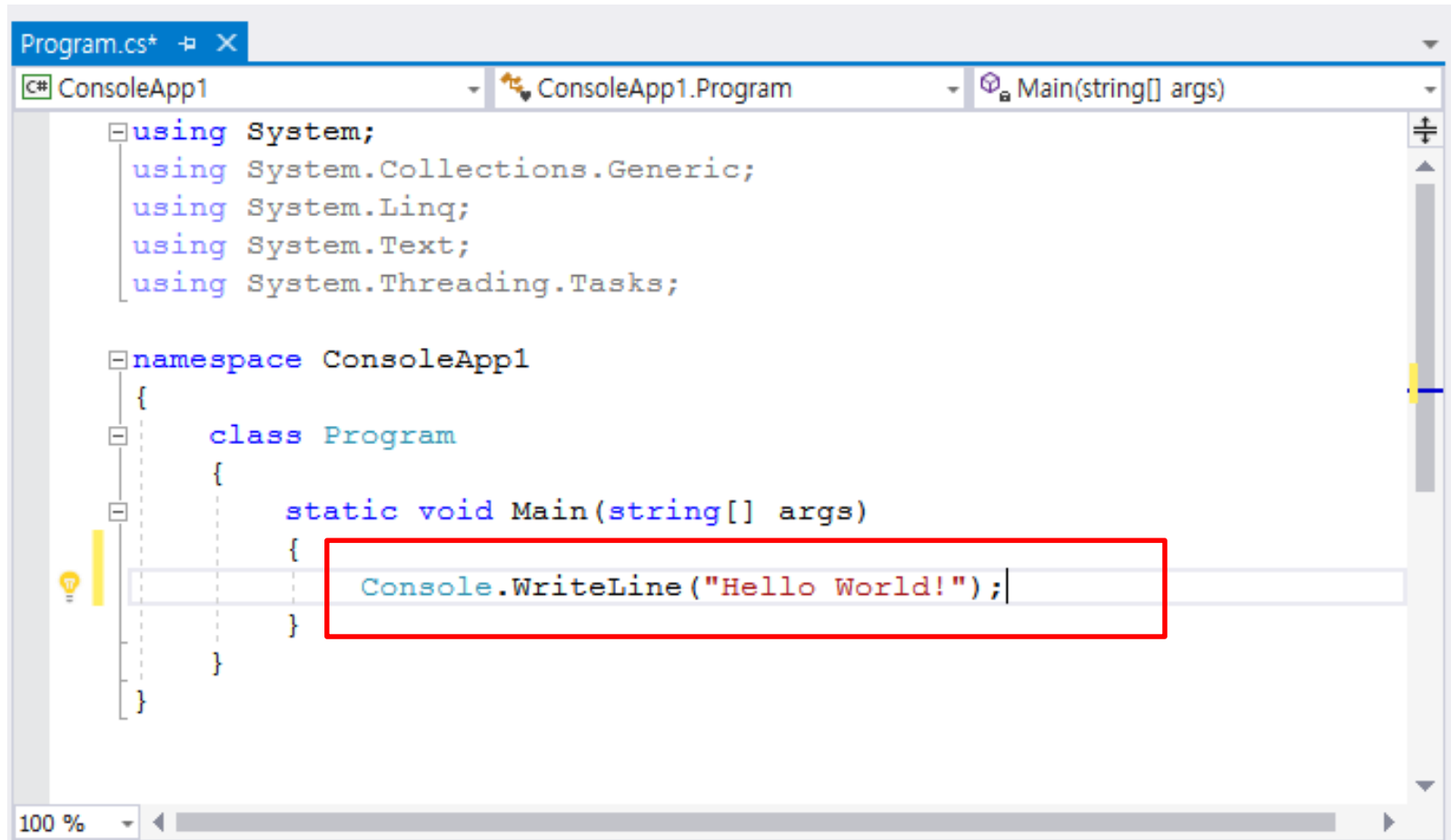
1. Hello world

- 프로그램 템플릿이 표시됨



1. Hello world

- 프로그램 작성

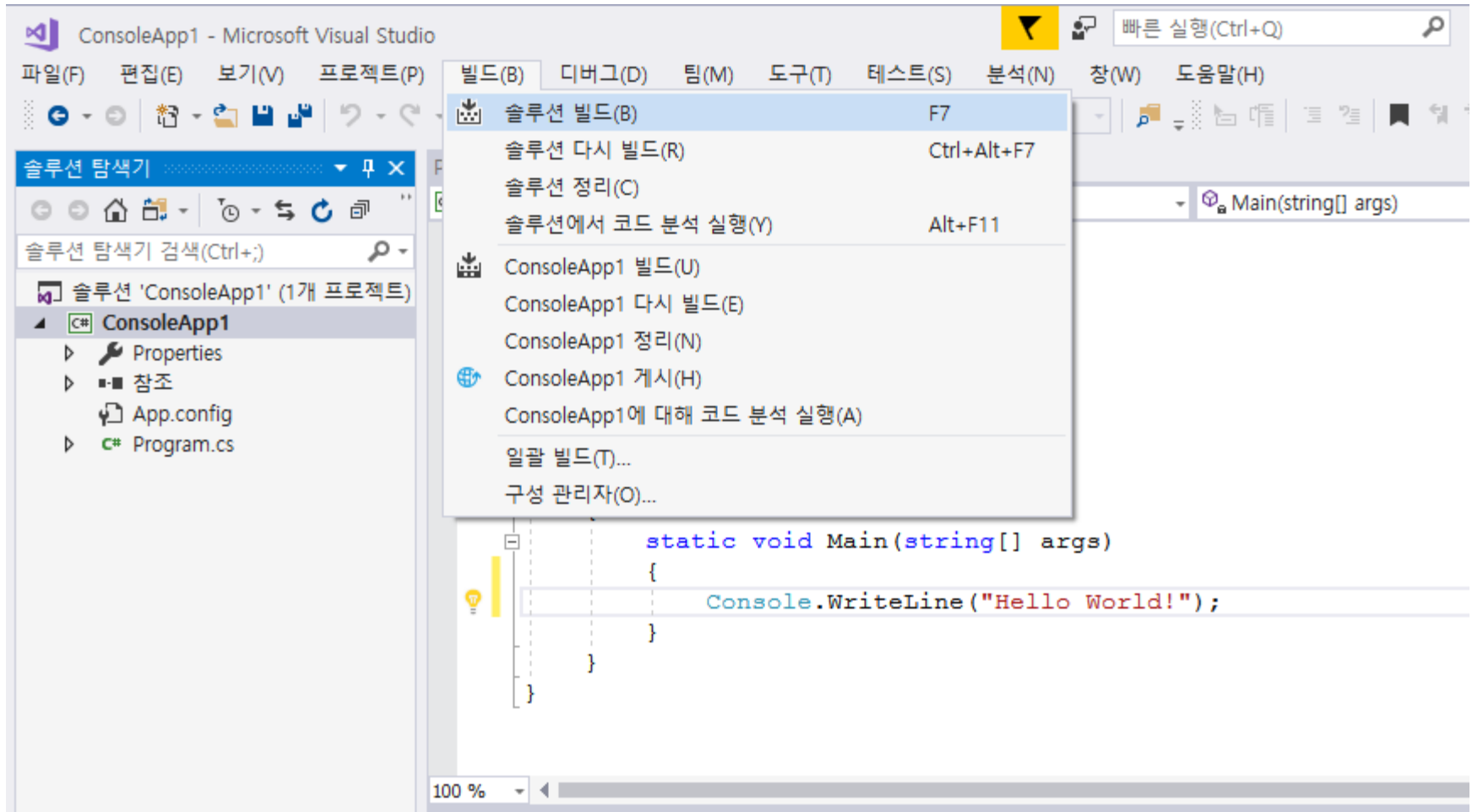


```
Program.cs* [X]
[C#] ConsoleApp1 ConsoleApp1.Program Main(string[] args)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

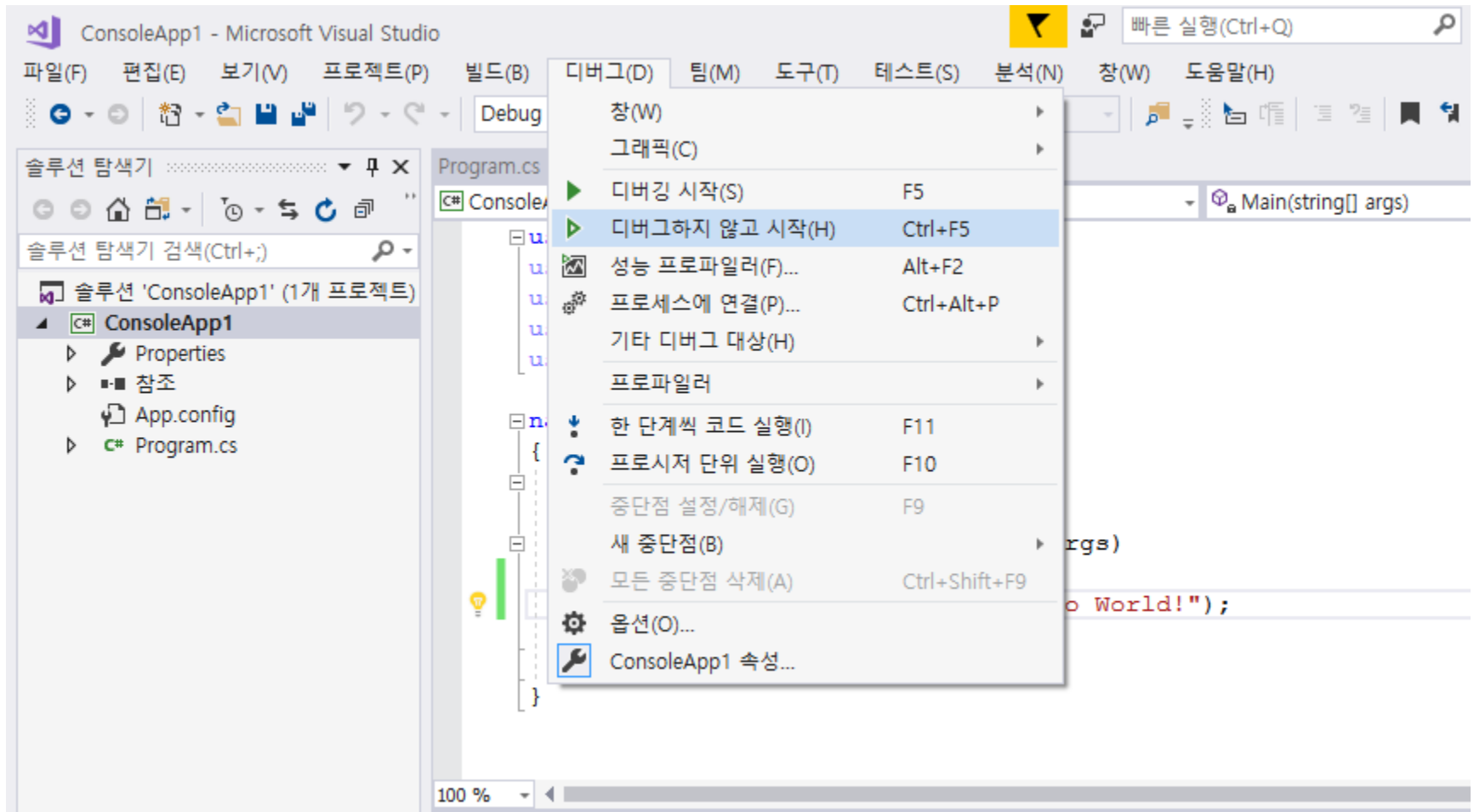
1. Hello world

- 솔루션 빌드



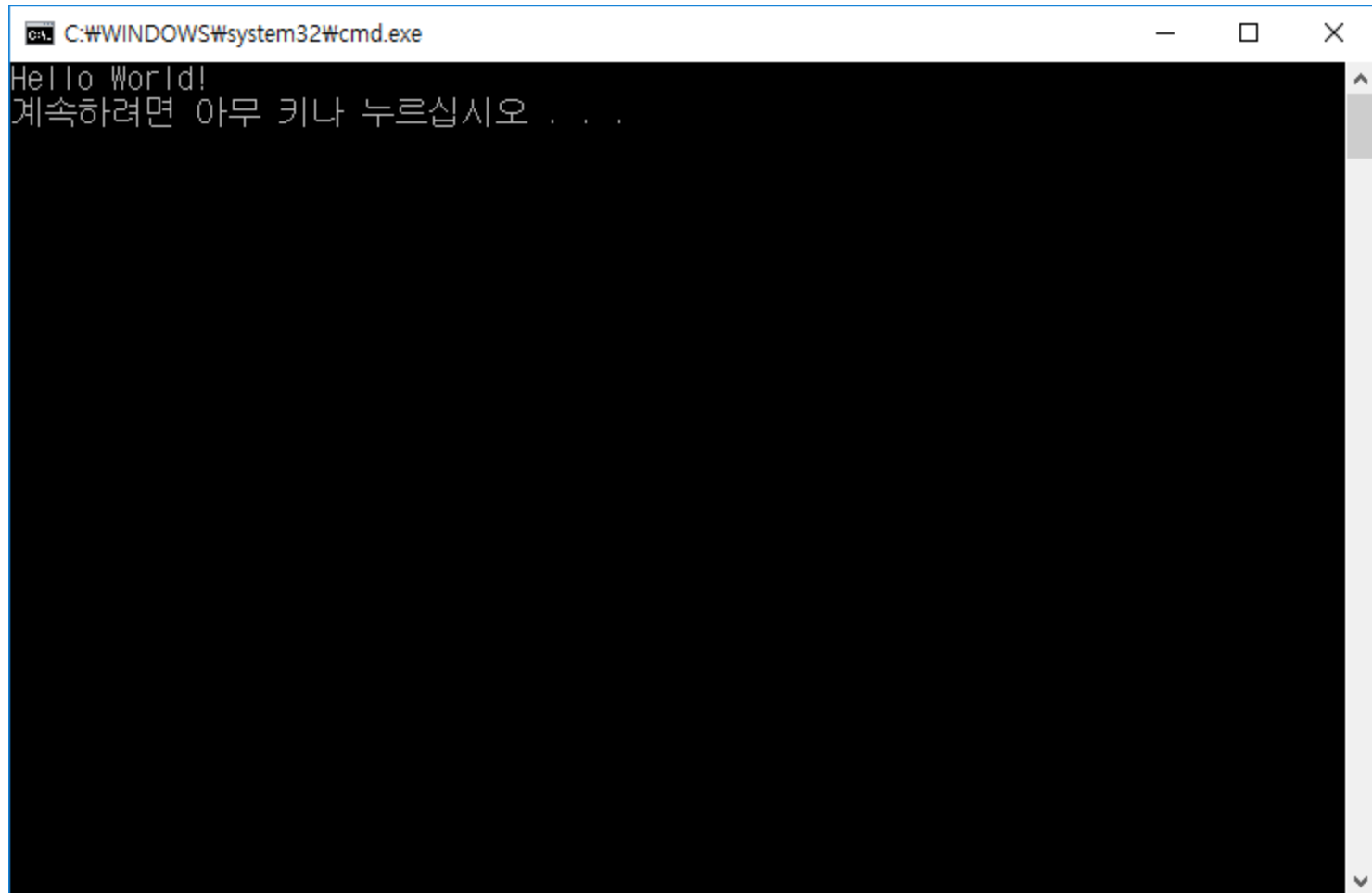
1. Hello world

- 프로그램 실행



1. Hello world

- 프로그램 실행 결과



A screenshot of a Windows command prompt window. The title bar at the top reads "C:\WINDOWS\system32\cmd.exe". The window has standard Windows window controls (minimize, maximize, close) on the right. The main area of the window is black with white text. The first line of text is "Hello World!". The second line of text is "계속하려면 아무 키나 누르십시오 . . .". A vertical scrollbar is visible on the right side of the window.

```
C:\WINDOWS\system32\cmd.exe
Hello World!
계속하려면 아무 키나 누르십시오 . . .
```

2. C# 프로그램 기초

- C# 프로그램 구조

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

2. C# 프로그램 기초

- namespace

서울시 강서구
부산시 강서구



- 이름의 충돌 사례
- 이를 해결하기 위한 방안

C++ 1동::마이클
C# 1동.마이클

2. C# 프로그램 기초

- namespace
 - 간단한 프로그램 작성시에는 namespace 지정을 생략 할 수 있다

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Hello World!");
    }
}
```

2. C# 프로그램 기초

- using
 - 제공되는 패키지(라이브러리)의 기능들을 가져다 프로그램 작성 가능
 - 패키지(라이브러리)마다 namespace 지정되어 있음
 - using : 어떤 패키지를 가져다 쓸지를 지정 함 (C 언어의 include)

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        Console.WriteLine("Hello World!");  
    }  
}
```

여기 있는 기능들은 가져다
쓴 것이 없으므로 생략 가능

2. C# 프로그램 기초

- using

```
class Program
{
    static void Main(string[] args)
    {
        System.Console.WriteLine("Hello World!");
    }
}
```

매번 namespace.xxx 와 같이 사용하는 것은 번거로움

using 사용

```
using System;

class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Hello World!");
    }
}
```

2. C# 프로그램 기초

- Class

- 문법

```
class 클래스명  
{  
    ...  
}
```

- 객체지향 프로그래밍의 핵심 개념 (2학년때 배움)
 - class 안에는 변수, 메소드(함수) 가 포함됨
 - 하나의 namespace는 여러 개의 클래스를 포함할 수 있음

```
System.Console.WriteLine("Hello World!");
```

namespace 이름 클래스 이름 method 이름

2. C# 프로그램 기초

- Main 메소드: 프로그램의 시작점
 - M은 반드시 대문자로 쓴다.
 - 반드시 public static void로 선언한다
 - 객체지향 프로그래밍에서는 함수를 메소드라고 부른다

```
using System;

class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Hello World!");
    }
}
```


2. C# 프로그램 기초

- Write/WriteLine 메소드
 - ◉ 스크린 모니터상에 정보를 보여준다. (C 에서 printf())
 - ◉ WriteLine은 캐리지 리턴 문자를 포함하여 출력 후에 다음 행으로 이동한다.
 - ◉ Write/WriteLine 메소드는 오버로드 되어있으므로 인자 형태로 숫자, 문자열등 여러 형태가 가능하다
- Read/ReadLine 메소드
 - ◉ 키보드로 값을 입력받을 때 사용한다.
 - ◉ Read 메소드는 키보드로부터 하나의 문자를 입력받는다. 반면 ReadLine 메소드는 한 줄을 입력받을 수 있다.

2. C# 프로그램 기초

- 주석문 (Comment)
 - 컴파일 시 무시되면 소스파일에서만 관리된다.
 - '//' 을 사용하여 한 줄을 주석처리 한다.
 - '/* */'를 사용하면 여러 라인을 주석처리 할 수 있다

```
using System;

// 메인 클래스
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Hello World!");
    }
}
```

2. C# 프로그램 기초

// example 2-1

```
using System;
```

```
class Program
```

```
{
```

```
    static void Main(string[] args)
```

```
    {
```

```
        int a =5, b= 10;
```

```
        Console.WriteLine("a+b="+ (a+b));
```

```
    }
```

```
}
```

C:\WINDOWS\system32\cmd.exe

a+b=15

계속하려면 아무 키나 누르십시오 . . .

2. C# 프로그램 기초

// example 2-2

```
using System;
```

```
class Program
```

```
{
```

```
    static void Main(string[] args)
```

```
    {
```

```
        int a,b;
```

```
        Console.Write("a값을 입력하세요: ");
```

```
        a = int.Parse(Console.ReadLine());
```

```
        Console.Write("b값을 입력하세요: ");
```

```
        b = int.Parse(Console.ReadLine());
```

```
        Console.WriteLine("a+b =" + (a+b) + " 입니다");
```

```
    }
```

```
}
```

C:\C:\WINDOWS\system32\cmd.exe

a값을 입력하세요: 10

b값을 입력하세요: 20

a+b =30 입니다

계속하려면 아무 키나 누르십시오 . . .

일반적인 C# 프로그램의 구조

프로그램

namespace

class

method

method

class

method

Main()

프로그램
시작지점

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
namespace Program
```

```
{
    class twodmatrix
    {
        int m, n;
        int[,] a;
        int[] b;
```

```
        twodmatrix(int x, int y) {
            m = x;
            n = y;
            a = new int[m, n];
            b = new int[m * n];
        }
```

```
        public void readmatrix() {
            for (int i = 0; i < m; i++)
            {
                for (int j = 0; j < n; j++)
                {
                    Console.WriteLine("a[{0},{1}]= ", i, j);
                    a[i, j] = Convert.ToInt32(Console.ReadLine());
                }
            }
        }
```

```
        public void printd() {
            for (int i = 0; i < m; i++)
            {
                for (int j = 0; j < n; j++)
                {
                    Console.Write("{0}\t", a[i, j]);

                }
                Console.WriteLine("\n");
            }
        }
```

```
        public void convert() {
            int k = 0;
            for (int i = 0; i < m; i++)
            {
                for (int j = 0; j < n; j++)
                {
                    b[k++] = a[i, j];
                }
            }
        }
```

```
        public void printoned()
        {
            for (int i = 0; i < m * n; i++)
            {
                Console.WriteLine("{0}\t", b[i]);
            }
        }
```

```
        public static void Main(string[] args) {
            twodmatrix obj = new twodmatrix(2,3);
            Console.WriteLine("Enter the Elements : ");
            obj.readmatrix();
            Console.WriteLine("\t\t Given 2-D Array(Matrix) is : ");
            obj.printd();
            obj.convert();
            Console.WriteLine("\t\t Converted 1-D Array is : ");
            obj.printoned();
            Console.ReadLine();
        }
    }
}
```

[연습 1]

- 1. 화면에서 두개의 숫자를 입력 받아 둘 중 큰 수를 출력하는 프로그램을 작성하시오 (if 문 사용 필요)

```
첫번째 숫자를 입력하세요: 5  
두번째 숫자를 입력하세요: 8  
두수중 큰 수는 8 입니다
```

- 2. . 화면에서 두개의 숫자를 입력 받아 큰 수, 작은 수의 순서로 출력하는 프로그램을 작성하시오 (if 문 사용 필요)

```
첫번째 숫자를 입력하세요: 4  
두번째 숫자를 입력하세요: 9  
크기의 순서는 9, 4 입니다
```