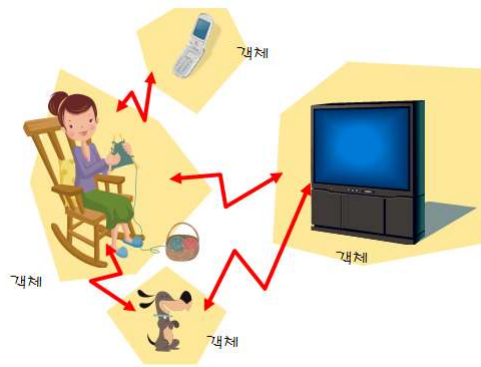


*C++ Espresso*

## 제1장 기초 사항



## 이번 장에서 학습할 내용

- C++ 언어의 역사
- C++ 언어의 특징
- 객체 지향의 간단한 소개
- C++ 개발 과정
- 첫번째 예제 설명
- 변수와 상수
- 연산자

C++에 대한  
기초적인  
사항들을  
살펴봅니다.

## 프로그래밍 언어

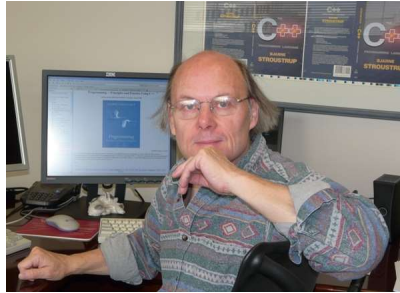
- C
  - 1972년경, 유닉스(Unix) 운영체제를 개발하기 위한 시스템 프로그래밍 언어
  - 하드웨어에 대한 보다 확실한 통제가 가능
  - 풍부한 연산자와 데이터 형(data type)을 갖고 있기 때문에 범용 프로그래밍 언어로서 널리 보급
  - 아직도 은행 전산실에서 많이 사용

## 프로그래밍 언어

- C++
  - C 언어는 1983년 프로그래밍 언어 C++로 발전
  - C 언어의 객체지향 버전(C + 1 → C++)
  - C++은 객체지향 프로그래밍(OOP Object-Oriented Programming)을 지원하기 위해
  - C 언어가 가지는 장점을 그대로 계승하면서 클래스, 상속성(inheritance) 등의 개념을 추가한 효과적인 언어
  - 포토샵, MS 윈도우, MS 오피스, 스타 크래프트 등 수많은 상용 소프트웨어들이 이 언어로 작성됨.
  - “손님이 과일장수에게 사과2개를 구입한다” 상황을 쉽게 구현 가능 → C에서는 어려움

## C++ 언어의 개발

- C++는 1980년대 초에 AT&T 벨연구소의 Bjarne Stroustrup에 의하여 개발
- C++는 C언어를 유지, 확장한 것 → C with Classes → C++
- C + 1 → C++
- C++는 C언어에 클래스 개념을 추가하고 이어서 가상 함수, 연산자 중괄호 정의, 다중 상속, 템플릿, 예외 처리 등이 기능이 차례로 추가



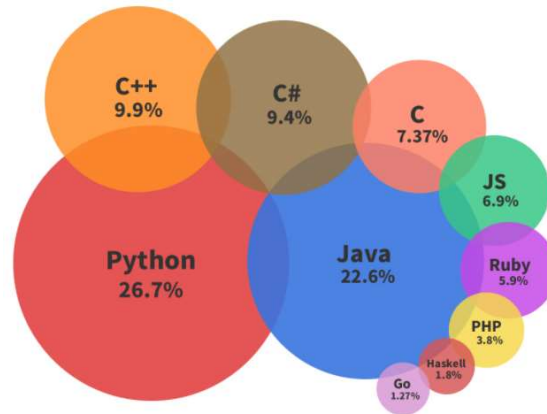
## 프로그래밍 언어

++ → #  
++

- C# ← (C++) +1
  - 마이크로소프트가 1990년대 말 공개된 언어
  - C#이라는 이름은 C++를 계승한다는 의미로 붙여졌음.
    - 그러나 C#은 C/C++과는 여러 가지 차이가 존재
    - C와 C++는 AT&T에서 개발 vs C#은 마이크로소프트에서 개발
  - 윈도우, 맥, 리눅스, 안드로이드, 아이폰 등의 다양한 운영체제나 플랫폼에서 동작하는 프로그램 개발
  - But java 많이 사용 !!

## 프로그래밍 언어 사용 순위

- 2016년 초에 , 코드 경진대회로 유명한 코드이벨(codeeval)에서
- 자사의 100만명이 넘는 회원들의 사용 통계에 기반해 발표한 프로그래밍 언어의 인기에 대한 통계



## 프로그래밍 언어 사용 순위

- IEEE Spectrum에서 조사한 순위(2020)
- 기업

일반 트렌드/선호

Language Ranking: **Jobs**

Rank	Language	Type
1	Python	🌐 🖥️ 📱
2	C	📱 🖥️ 🌐
3	Java	🌐 📱 🖥️
4	Go	🌐 🖥️
5	C++	📱 🖥️ 🌐
6	JavaScript	🌐

Language Ranking: **Trending**

Rank	Language	Type
1	Python	🌐 🖥️ 📱
2	Java	🌐 📱 🖥️
3	C	📱 🖥️ 🌐
4	C++	📱 🖥️ 🌐
5	Go	🌐 🖥️
6	JavaScript	🌐

## 프로그래밍 언어 사용 순위

- 구인 구직 사이트, 사람인, <https://www.saramin.co.kr>
- 메뉴에서 채용정보 → 직업별 → IT 개발.데이터 → 기술스택(광고 많은 순)

Java(2,947) C++(2,923) Javascript(1,962) Python(1,937) Linux(1,911) AWS(1,624) SQL(1,568) C#(1,534) C언어(1,364)

Spring(1,354) MySQL(1,351) Git(1,337) HTML(1,226) React(1,216) CSS(1,030) R(901) Android(815) SpringBoot(769)

iOS(692) Node.js(691) ECMAScript(644) Docker(642) MSSQL(630) jQuery(622) JSP(617) TypeScript(593) HTML5(592)

## 통합 개발 환경의 종류

- 비주얼 studio (Visual Studio)
  - 윈도우 기반의 거의 모든 형태의 응용 프로그램 제작 가능
  - 책에서 사용하는 버전: 비주얼 studio 2015 ??
  - 수업자료에서는 비주얼 studio Community 2019
    - <https://visualstudio.microsoft.com/ko/vs/older-downloads/>
    - 최신버전 사용도 가능 ← 과제는 소스파일만 제출
- 통합 개발 환경 (IDE: integrated development environment):
  - 에디터 + 컴파일러 + 디버거
  - 옛날에는 모두 분리

## VS 2019 설치

- <https://visualstudio.microsoft.com/ko/vs/older-downloads/>

### 최신 Visual Studio를 사용해 보셨나요?

최신 64비트 Visual Studio 2022를 사용하여 적합한 IDE를 만들고, 더 스마트한 앱을 빌드하고, 클라우드와 통합하고, 성능을 최적화하여 경쟁력을 유지하세요.

Visual Studio 2022 다운로드 Mac용 Visual Studio 2022 다운로드

Visual Studio 2022의 새로운 기능 확인 >

### 아직 이전 버전이 필요하신가요?

아래에서 제품을 선택하고 다운로드 단추를 클릭하여 Visual Studio(MSDN) 구독에 로그인하거나 무료 Dev Essentials 프로그램에 가입하여 이전 버전에 액세스하세요.

모두 확장 모두 축소

> 2019

> 2017

> 2015

> 2013

## VS 2019 설치

모두 확장

2019

### Visual Studio 2019 및 기타 제품

다음 목록의 제품을 다운로드하려면 다운로드 단추를 클릭하고 메시지가 표시되면 Visual Studio 구독 계정으로 로그인하세요. Visual Studio 구독이 없는 경우 로그인 페이지에서 "새 Microsoft 계정 만들기"를 클릭하여 무료로 계정을 하나 만들 수 있습니다.

Visual Studio Professional 2019; Visual Studio Enterprise 2019;

Mac용 Visual Studio 2019

Visual Studio 2019용 빌드 도구

Visual Studio Team Explorer 2019

Visual Studio 2019용 에이전트

Visual Studio 2019용 IntelliTrace Standalone Collector

Visual Studio 2019용 성능 도구

Visual Studio 2019용 원격 도구

Visual Studio 2019용 Microsoft Visual C++ 재배포 가능 패키지

다운로드

## VS 2019 설치, Visual Studio Community 2019

Product	Version	Release Date	Language	Architecture	Type	Info	Keys	Download
설치시 데스크 탑용 체크하고 설치								
Visual Studio Enterprise 2019	16.11	2023. 8. 8.	Multiple Languages	x86	exe	0		Download ↓
Visual Studio Team Explorer 2019	16.11	2023. 8. 8.	Multiple Languages	x86	exe	0		Download ↓
Visual Studio Professional 2019	16.11	2023. 8. 8.	Multiple Languages	x86	exe	0		Download ↓
Visual Studio Community 2019	16.11	2023. 8. 8.	Multiple Languages	x86	exe	0		Download ↓
Agents for Visual Studio 2019	16.11	2023. 8. 8.	English	x64	exe	0		Download ↓
Build Tools for Visual Studio 2019	16.11	2023. 8. 8.	Multiple Languages	x86	exe	0		Download ↓

**Community 설치**

## 프로젝트 생성하기

### Visual Studio 2019

최근 파일 열기(R)

API.sln  
D:\USR\수업\APIW2019 APIW2019 ppt\4187\wcode

12장\_1절.sln  
D:\USR\수업\OpenCV 2020\교재예제\w12장\_예제\w12장\_1절

11장\_3절.sln  
D:\USR\수업\OpenCV 2020\교재예제\w11장\_예제\w11장\_3절

TextExam.sln  
C:\Users\admin\Desktop\vs 2019 opencv\TextExam

11장\_2절.sln  
D:\USR\수업\OpenCV 2020\교재예제\w11장\_예제\w11장\_2절

06장\_3절.sln  
D:\USR\수업\OpenCV 2020\교재예제\w06장\_예제\w06장\_3절

06장\_예제.sln  
D:\USR\수업\OpenCV 2020\교재예제\w06장\_예제\w06장\_1절

### 시작

코드 복제 또는 체크 아웃(C)  
GitHub 또는 Azure DevOps 같은 온라인 리포지토리에서 코드 가져오기

프로젝트 또는 솔루션 열기(P)  
로컬 Visual Studio 프로젝트 또는 sln 파일 열기

로컬 폴더 열기(F)  
폴더 내에서 탐색 및 코드 편집

**새 프로젝트 만들기(N)**  
시작하려면 코드 스캐폴딩과 함께 프로젝트 템플릿을 선택하세요.

코드를 사용하지 않고 계속(W) →

## 프로젝트 생성하기

### 새 프로젝트 만들기

최근 프로젝트 템플릿(R)

빈 프로젝트 C++

콘솔 앱 C++

빈 프로젝트  
Windows용 C++를 사용하여 처음부터 시작합니다. 시작 파일을 제공하지 않습니다.  
C++ Windows 콘솔

콘솔 앱  
Windows 터미널에서 코드를 실행합니다. 기본적으로 "Hello World"를 출력합니다.  
C++ Windows 콘솔

Windows 데스크톱 마법사  
마법사를 사용하여 고유한 Windows 앱을 만드세요.  
C++ Windows 데스크톱 콘솔 라이브러리

Windows 데스크톱 애플리케이션  
Windows에서 실행되는 그래픽 사용자 인터페이스를 사용하는 애플리케이션용 프로젝트입니다.  
C++ Windows 데스크톱

공유 항목 프로젝트  
공유 항목 프로젝트는 여러 프로젝트 간에 파일을 공유하는 데 사용됩니다.  
C++ Windows Android iOS Linux 데스크톱 콘솔  
라이브러리 UWP 게임 모바일

뒤로(B) 다음(N)

## 프로젝트 생성하기

### 새 프로젝트 구성

빈 프로젝트 C++ Windows 콘솔

프로젝트 이름(N)  
Test\_10

위치(L)  
C:\Users\wadmin\Desktop\프로그래밍실습

솔루션 이름(M)  
Test\_10

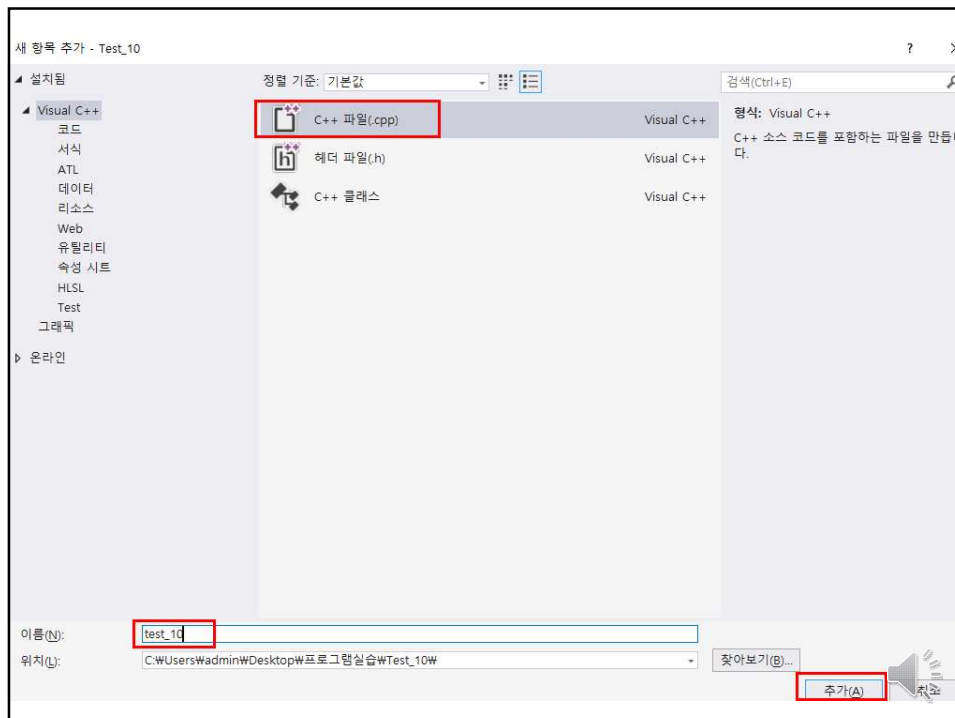
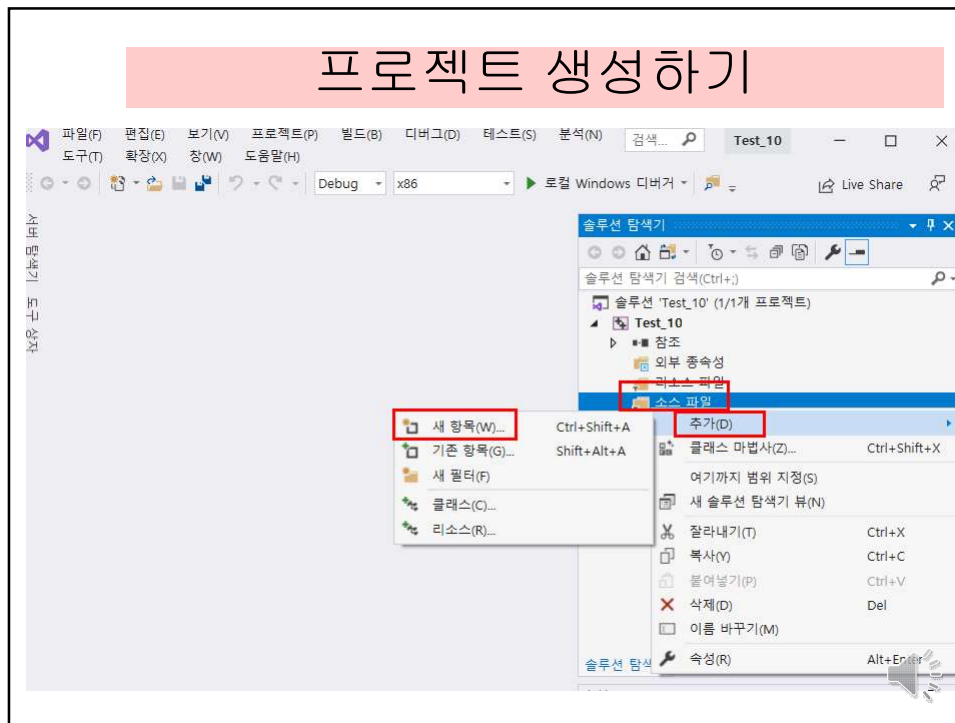
☒ 솔루션 및 프로젝트를 같은 디렉터리에 배치(D)

바탕화면에 각자 학번으로  
폴더 만들고 지정

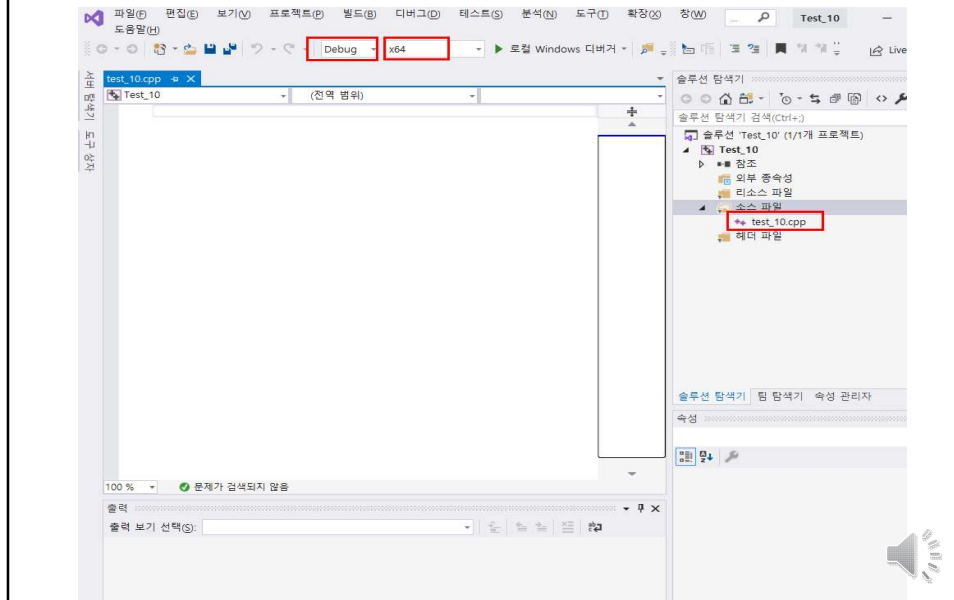
뒤로(B) 만들기(M)



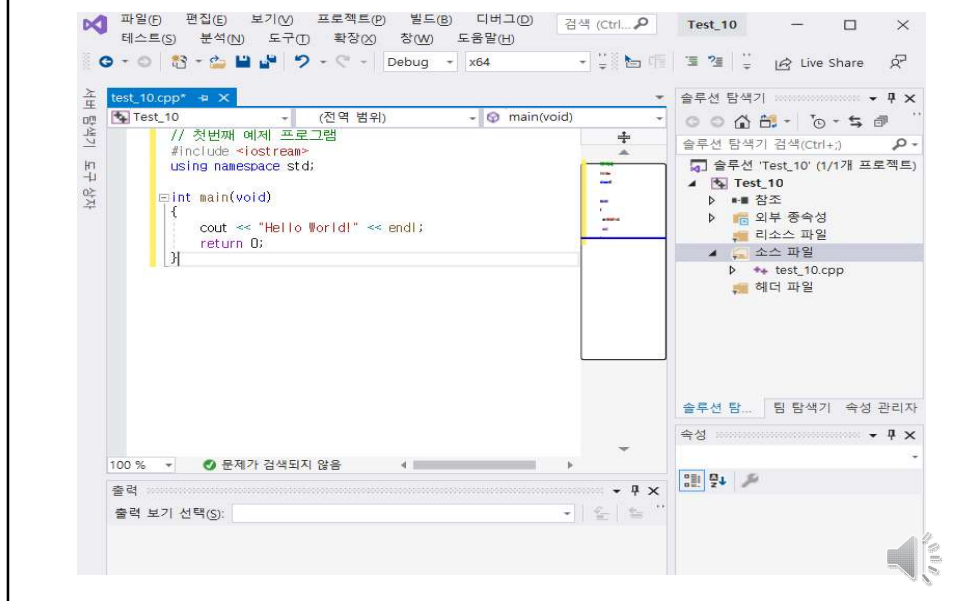
## 프로젝트 생성하기



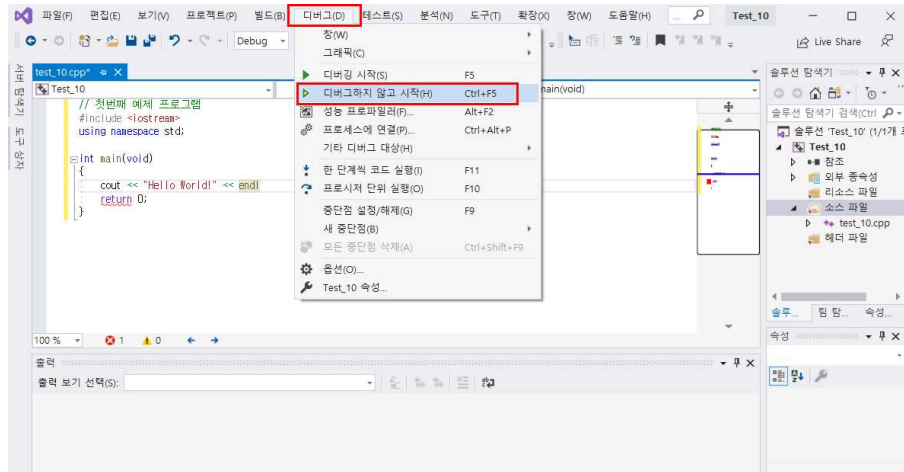
## 프로젝트 생성하기



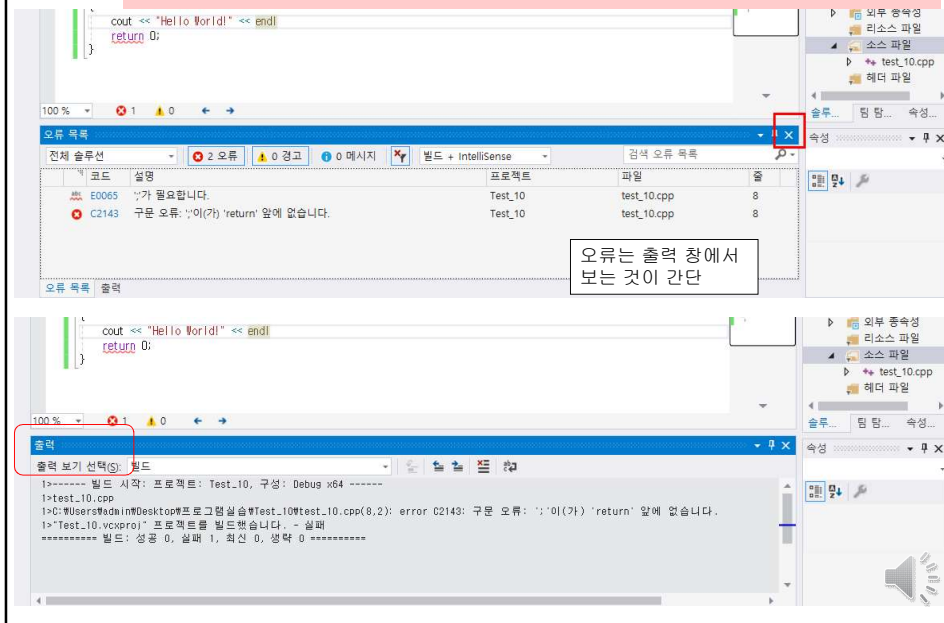
## 프로젝트 생성하기



## 프로젝트 생성하기



## 프로젝트 생성하기



## cout, cin

- cout → printf 와 동일 기능, 화면에 출력시 사용
- printf( ) 에서는 %d, %f 등을 사용하여 출력 → 복잡
- printf 이용 출력 방식

```
int x=10;          float y = 4.5;
printf(' %d 는 %f 보다 크다.\n', x, y);
```

- cout 이용 출력 방식 → "<<" 이용하여 변수, 문자열 등을 출력

```
int x=10;          float y = 4.5;
cout << x << "는 " << y << "보다 크다." << endl;
```

- \n → endl 로 대체
- 입력문 : scanf → cin 으로 사용

## cout, cin

- cin → scanf 와 동일 기능, 키보드에서 입력 받음
- scanf( ) 에서는 %d, %f 등을 사용하여 출력 → 복잡

```
#include <iostream>
```

```
int main() {
    string name;
    int age;

    cout << "이름을 입력하세요: ";
    cin >> name;

    cout << "나이를 입력하세요: ";
    cin >> age;

    cout << "안녕하세요, " << name << "님! " << age << "살이군요." << endl;

    return 0;
}
```

## 첫 번째 프로그램의 분석

C++ 언어에서는 대문자와 소문자를 구별

test.cpp

// 첫번째 예제 프로그램

#include <iostream>

using namespace std;

int main(void)

{

cout << "Hello World!" << endl;

return 0;

}

실행결과

Hello World!

주석

헤더파일

이름공간 설정

화면에 문자열 출력



## 헤더 파일 포함

#include <iostream>

- #include는 소스 코드 안에 특정 파일을 포함시킴
- 헤더 파일(header file): 컴파일러가 필요로 하는 정보(함수 정의, type, 클래스 정의)를 가지고 있는 파일
  - 특정 함수/type/class를 사용하려면 필요한 헤더파일을 include 해야함
- iostream: 표준 입출력 스트림
- 주의! : 전처리기 지시자 문장 끝에는 세미콜론을 붙이면 안 된다.

#include <iostream>  
...  
HelloWorld.cpp

// iostream ...  
#pragma once  
...  
iostream

iostream은 입출력에 관한 클래스를 가지고 있는 헤더 파일입니다.



## 이름 공간

`using namespace std;`

- namespace(이름공간)
  - 한 프로젝트를 철수, 순이 2명이 나누어 작업
  - 각각이 같은 이름 함수 생성 사용하면 두 사람 작업 합치는 경우 문제 발생
  - 철수와 순이가 함수 `sum()` 를 각각 작성
    - `sum()` 을 호출하면 → 누구의 `sum()` 이 호출?
    - 철수의 `sum()` 을 호출하려면? → 해결 by namespace
- 실제 작업시
  - 각 사람이 각각의 namespace 정의 한 후 그 안에 함수들 작성하고
  - 호출시 각 namespace 지정하고 함수 호출

함수 작성시

```
namespace BestC {
    sum()
    ...
}
```

호출시

```
BestC:: sum()
```

```
namespace ProgC {
    sum()
    ...
}
```

```
ProgC:: sum()
```



## std::cout, std::cin, std::endl

```
#include <iostream>
using namespace std;

int main(void)
{
    int num=20;
    cout<<"Hello World!"<<endl;
    cout<<"Hello " <<"World!"<<endl;
    cout<<num<< ' ' <<'A';
    cout<< ' ' <<3.14<<endl;
    return 0;
}
```

- `<iostream>`에 정의가 있는 `cout`, `cin` 그리고 `endl` 은 이름공간 `std` 안에 선언.
- 이름공간 `std`에 선언된 것은 `std`라는 이름공간의 사용 없이 접근하겠다는 내용  
→ 이름공간 지정 없으면 `std::cout` 과 같이 사용해야 함

std::cout → "이름공간 std에 선언된 cout"  
std::cin → "이름공간 std에 선언된 cin"  
std::endl → "이름공간 std에 선언된 endl"

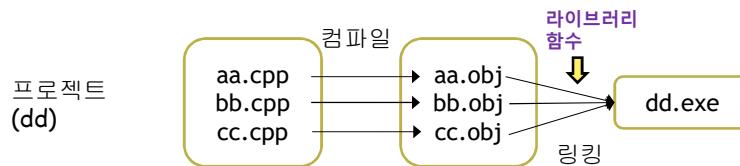


```
namespace std
{
    cout . . . .
    cin . . . .
    endl . . . .
}
```



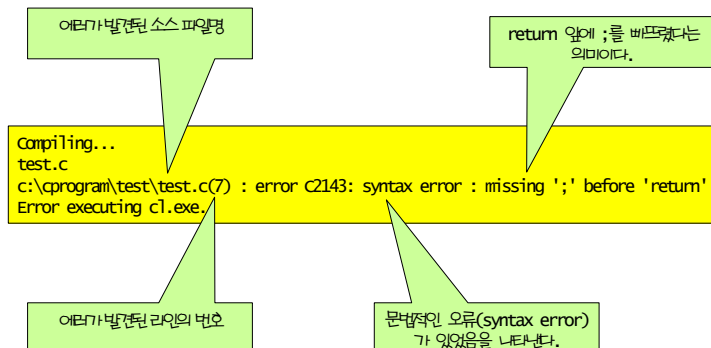
## 프로그래밍 실행 과정

1. 소스 프로그램(~.cpp) 작성
  - 편집기(에디터)를 사용하여 문법에 맞게 파일에 작성
2. 컴파일(Compile)
  - 소스 프로그램에 문법적 오류가 있는지 검사
  - 기계어로 번역, 목적 파일(~.obj) 생성
3. 링킹(Linking)
  - 여러 목적 파일을 묶어 하나의 실행 파일(~.exe)로 만들어 주는 기능
4. 실행



## 오류

1. 컴파일 오류 → 문법 오류(syntax error)
2. 링크 오류 → 작성한 함수 등이 존재하지 않는 경우.
3. 실행 오류 : 결과가 다른 경우



## 링크 오류(error LNKxxxx)

```
#include <iostream>
#include <string>
using namespace std;

int sum(int);

int main() {
    int a=5;
    cout << a;
    cout << sum(a);
}
```

오류 있는 곳의 라인 번호 없음  
호출되는 함수가 없는 경우 발생  
sum(...) 함수가 없음

실행 보기 선택(S): 빌드

1>----- 모두 다시 빌드 시작: 프로젝트: Exam, 구성: Debug Win32 -----

1> ExamReport.cpp

1>ExamReport.obj : error LNK2019: "int \_\_cdecl sum(int)" (?sum@@YAH@Z) 외부 기호(참조 위치: \_main 함수)에서 확인하지 못했습니다.

1>0: #강의자료#강의결과물#수업 성적 등#2015 강의#2015 1 학기#작업제출#ExamDebug#Exam.exe : fatal error LNK1120: 1개의 확인할 수 없는 외부 참조입니다.

===== 모두 다시 빌드: 성공 0, 실패 1, 생략 0 =====

## 문자열 타입

- string 타입을 제공한다. : 문자열을 저장, 처리하기 위한 type

string.cpp

```
#include <iostream>
#include <string>
using namespace std;

int main(void)
{
    string s1 = "Good";
    string s2 = "Morning";
    string s3 = s1 + " " + s2 + "!\n";
    cout << s3;
    return 0;
}
```

이 헤더파일을 반드시 포함하여야 한다.

+ 연산자 : 문자열 연결

실행결과

Good Morning!



## 문자열 타입

- 문자열의 비교는 == 연산자로 가능

```
int main(void)
{
    string s1 = "Good";
    if( s1 == "Good" ){
        ...
    }
    else if( s1 == "Bad" ){
        ...
    }
    return 0;
}
```

## 예제

cin.cpp

```
#include <iostream>
#include <string>
using namespace std;

int main(void)
{
    string name;
    cout << "이름을 입력하시오: ";
    cin >> name;
    cout << name << "을 환영합니다." << endl;
    return 0;
}
```

문자열이 name으로 입력된다.

C++ 입출력 함수

- cout << 변수(or 상수);
- cin >> 변수;

실행결과

이름을 입력하시오: 홍길동  
홍길동을 환영합니다.

## 기호 상수의 장점

상수가 사용된 모든 곳을 변경하여야 한다.

```
income = salary - 0.15 * salary;
...
expenditure += 0.15 * salary;
```

기호상수의 정의만 바꾸면 된다.

```
#define TAX_RATE 0.15
income = salary - TAX_RATE * salary;
...
expenditure += TAX_RATE * salary;
```

## 기호 상수를 만드는 방법

### const 키워드 이용

```
#include <iostream>
using namespace std;    // 이름공간설정
int main()
{
    const int MONTHS = 12; // 기호상수선언, MONTHS 는 값 변경 금지
    double m_salary, y_salary; // 변수선언

    cout << "월급을 입력하시요: "; // 입력 안내문
    cin >> m_salary;

    y_salary = MONTHS * m_salary; // 순수입 계산
    cout << "연봉은" << y_salary << "입니다" << endl;
    return 0;
}
```

## 나머지 연산자



```
#include <iostream>
using namespace std;
int main()
{
    int input, minute, second;
    const int SEC_PER_MINUTE=60;

    cout << "초단위의 시간을 입력하시요:(32억초 이하) ";
    cin >> input;          // 초 단위의 시간을 읽는다.

    minute = input / SEC_PER_MINUTE; // 몇분, 나누기(몫)
    second = input % SEC_PER_MINUTE; // 몇초, 나머지 연산자

    cout << input << "초는" << minute << "분" << second << "초입니다." << endl;

    return 0;
}
```

10 / 3 = 3  
10 % 3 = 1



초단위의 시간을 입력하시요:(32억초이하) 1000  
1000초는 16분 40초 입니다.

## Q & A

