제 1 강

C++언어의 소개

C++언어의 소개

1 C++ 언어의 개요

2 C++ 프로그램의 작성 및 빌드

3 C++ 프로그래밍 첫걸음

• 1. C++ 언어의 개요

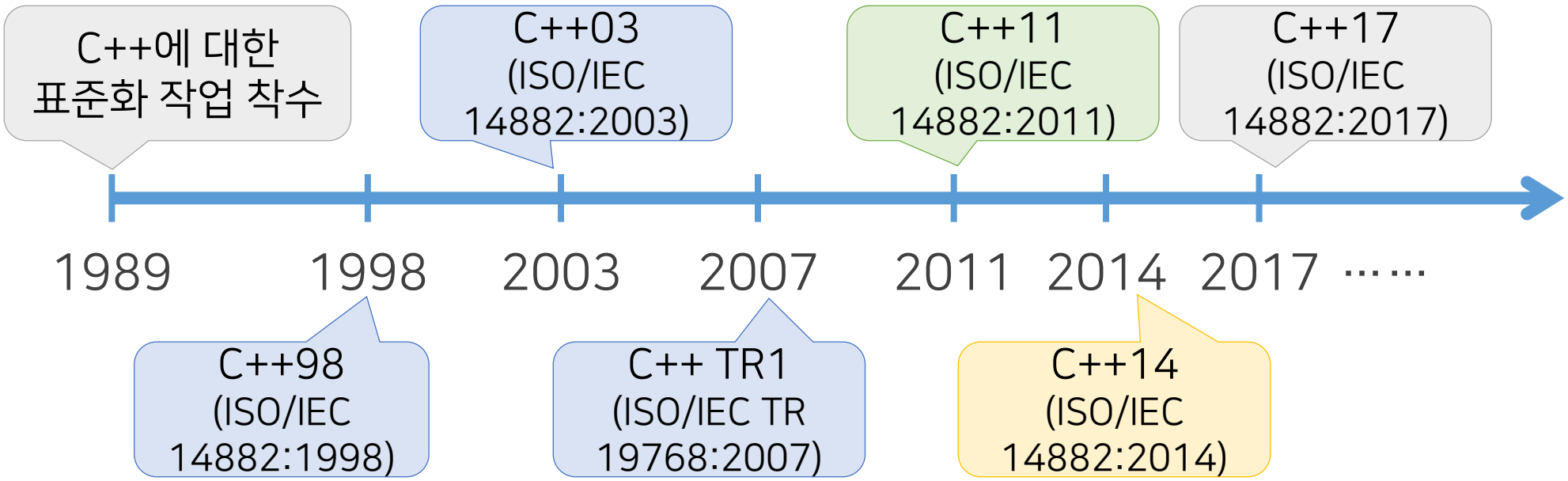
• (1) C와 C++ 언어

• C++ 언어란?

- 1979년 Bell 연구소의 Bjarne Stroustrup이 C 언어를 확장하여 만든 프로그래밍 언어
  - C with Classes
  - ⇨ 1983년부터 C++라는 이름을 사용함
- 주요 확장 내용
  - 객체지향 프로그래밍 : 클래스, 상속, 다형성, 동적 바인딩 등
  - 일반화 프로그래밍 : 템플릿
  - 예외처리

• (2) C++ 언어의 표준

- 국제표준화기구(ISO)의 C++ 표준화 연혁



C++언어의 소개

1 C++ 언어의 개요

2 C++ 프로그램의 작성 및 빌드

3 C++ 프로그래밍 첫걸음

• (1) C++ 프로그램의 소스 파일 (1/2)

C++ 소스 프로그램 파일 | C++ 헤더파일

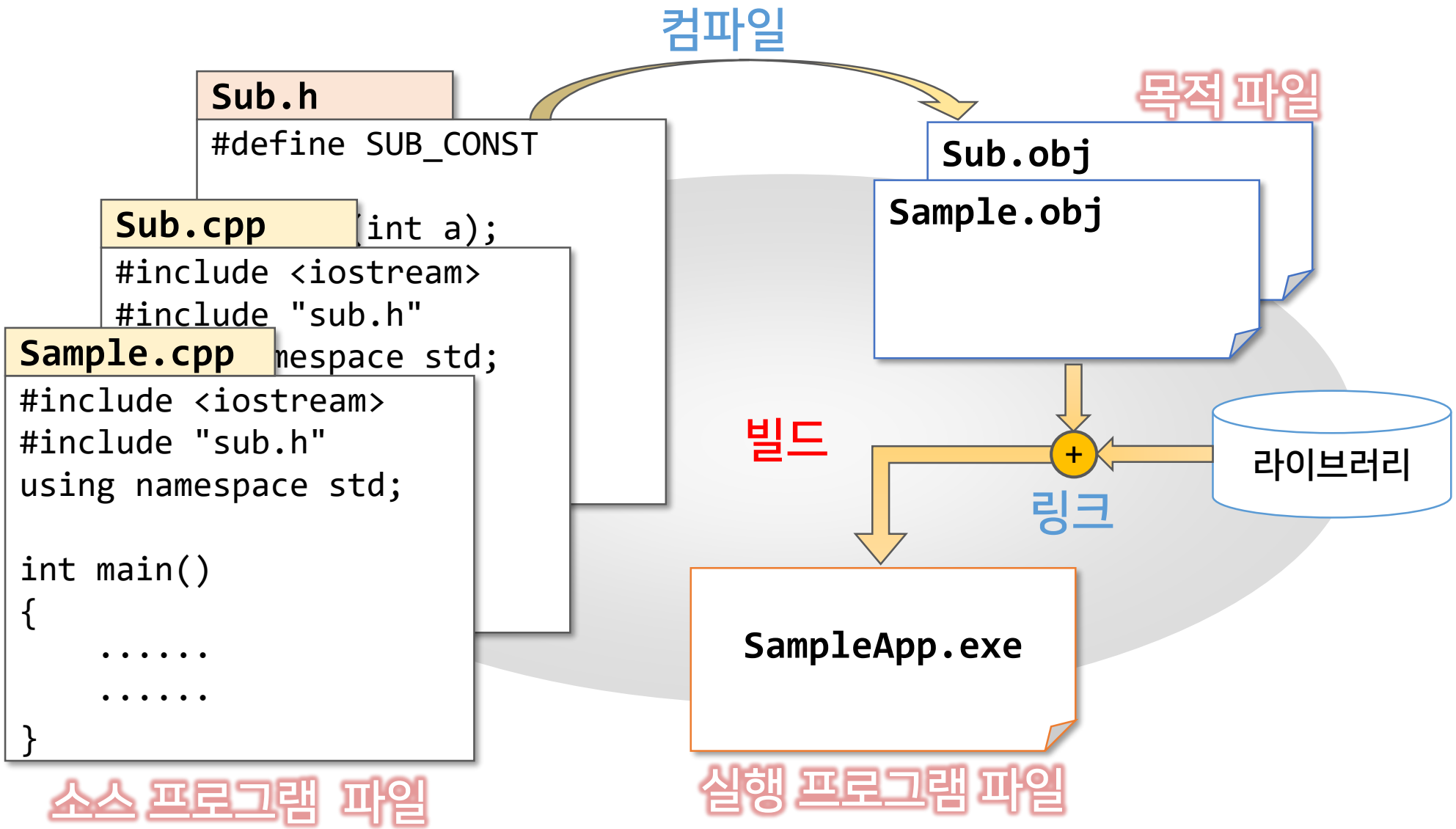
- 처리하고자 하는 작업을 수행하는 C++ 프로그램 명령어들을 담고 있는 파일
- 파일의 확장자 : '.cpp', '.cxx', '.C' 등을 사용함

• (1) C++ 프로그램의 소스 파일 (2/2)

C++ 소스 프로그램 파일 | C++ 헤더파일

- 클래스, 함수의 원형, 매크로, 전역변수, 상수 등 여러 소스 파일에 공통적으로 선언되는 내용을 담고 있는 파일
- 단독으로 컴파일되지 않고, #include라는 선행처리기 지시어에 의해 소스 프로그램 파일에 삽입되어 함께 컴파일됨
- 확장자 : 일반적으로 '.h'를 사용함

• (2) C++ 프로그램의 빌드 (1/2)



• (2) C++ 프로그램의 빌드 (2/2)

- 필요한 도구

  - 소스 프로그램 및 GUI 등의 편집기
  - 컴파일러
  - 링커
  - 디버깅 도구

→ 통합개발환경(IDE)

C++언어의 소개

1 C++ 언어의 개요

2 C++ 프로그램의 작성 및 빌드

3 C++ 프로그래밍 첫걸음

• (1) C++ 소스 프로그램

• FirstStep.cpp

```
1  #include <iostream>
2
3  int main()
4  {
5    // 표준 출력 스트림으로 문장을 출력함
6    std::cout << "나의 첫 번째 C++ 프로그램"
                 << std::endl;
7    return 0;
8  }
```

주석(comment)

• (2) 주석 (1/2)

- C++에서 주석(comment)을 작성하는 방법

1. '/*'와 '*/' 사이에 문장을 작성

```
/* a와 b의 값 중에서
   더 큰 값을 출력한다.  */
if (a > b)
   cout << a << endl;
else
   cout << b << endl;
```

• (2) 주석 (2/2)

- C++에서 주석(comment)을 작성하는 방법

2 '//'를 기입하면 그 행의 나머지는 주석임

```
// a와 b의 값 중에서
// 더 큰 값을 출력한다.
if (a > b)
    cout << a << endl;
else
    cout << b << endl;
```

• (3) 선행처리 (1/4)

• 선행처리의 지시

```
1  #include <iostream>
2
3  int main()
4  {
5    // 표준 출력 스트림으로 문장을 출력함
6    std::cout << "나의 첫 번째 C++ 프로그램"
                 << std::endl;
7    return 0;
8  }
```

선행처리기 지시어

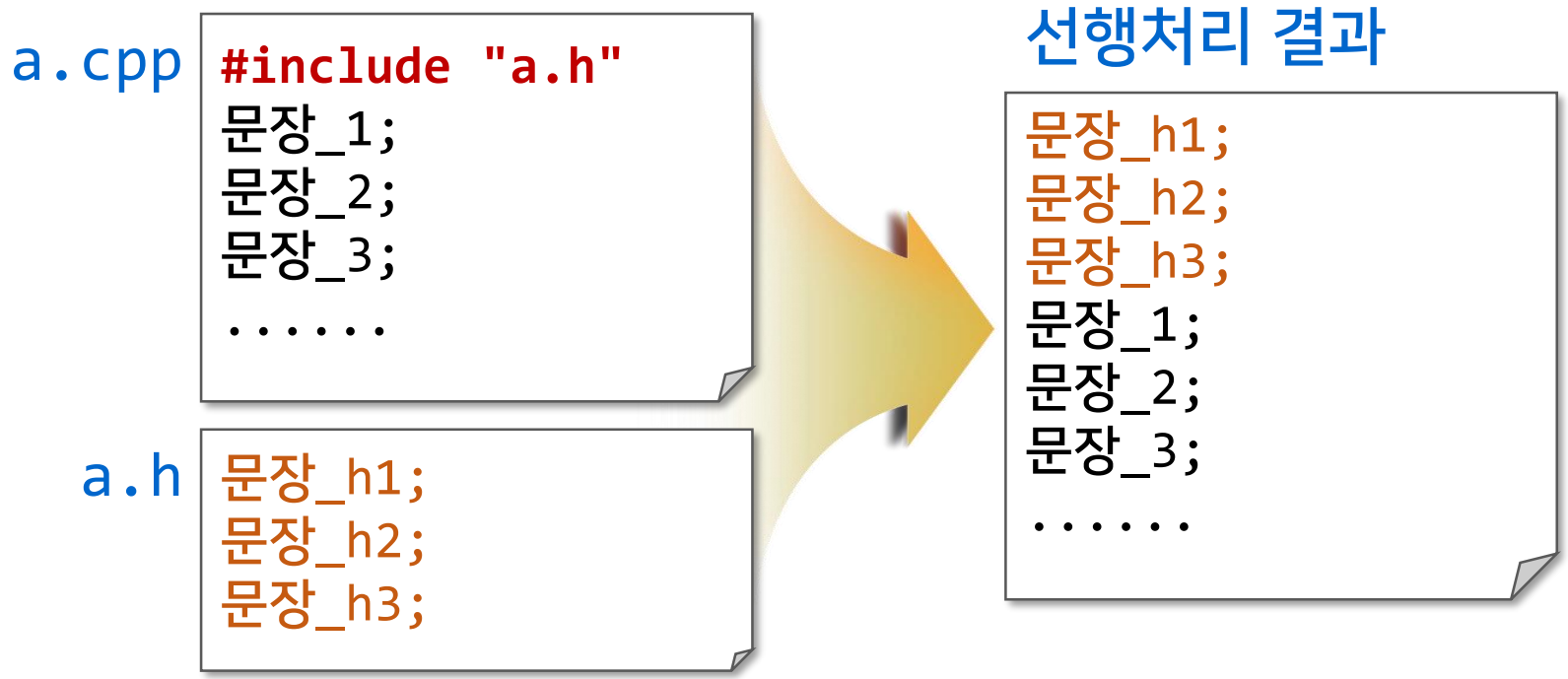
• (3) 선행처리 (2/4)

• 선행처리란?

- C++ 프로그램을 컴파일하기 전에 소스 프로그램을 가공하여 컴파일러가 실제로 번역할 소스 프로그램을 만드는 것
- 선행처리기 지시어(preprocessor directives)로 처리를 지시함
  - 선행처리기 지시어는 '#'로 시작함
  - 선행처리기 지시어 문장은 한 행에 한 개의 문장을 작성함
- 대표적인 선행처리
  - 헤더파일 삽입 : `#include`
  - 매크로 선언 및 해제 : `#define`, `#undef`
  - 조건부 컴파일 : `#if`, `#ifdef`, `#ifndef`

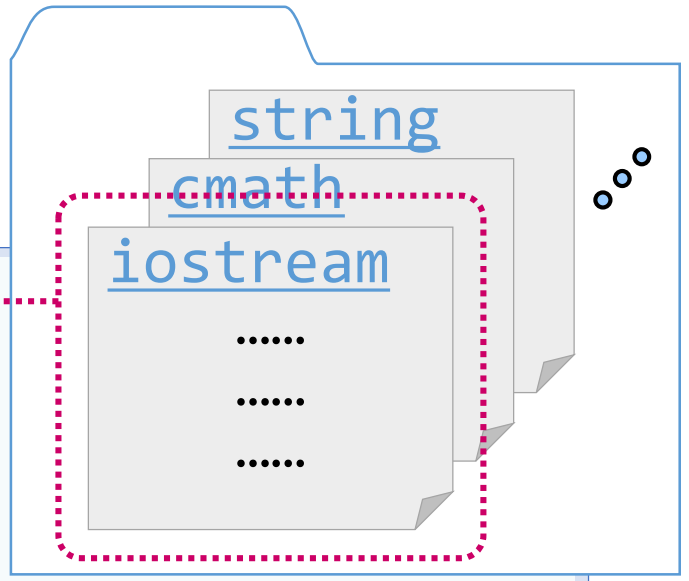
• (3) 선행처리 (3/4)

• 선행처리의 예 - 헤더파일 삽입



• (3) 선행처리 (4/4)

• 선행처리의 예 - 헤더파일




```
1  #include <iostream>
2
3  int main()
4  {
5    // 표준 출력 스트림으로 문장을 출력함
6    std::cout << "나의 첫 번째 C++ 프로그램"
                 << std::endl;
7    return 0;
8  }
```

• (4) 문장

• C++ 프로그램의 문장

- 하나의 문장은 단어와 연산자, 숫자, 문자, 문자열, 문장 부호, 빈칸 등을 정해진 문법에 따라 나열하여 작성함
- 문장의 끝에는 세미콜론(;)을 기입하여 다음 문장과 구분함

• 블록(block)

- 한 개 이상의 문장을 중괄호('{'와 '}') 안에 나열하여 묶어 놓은 것
- 여러 개의 문장을 묶어 하나의 문장처럼 취급하거나 함수의 몸체를 구성하기 위해 사용됨

• (5) 함수

• 함수의 구성

```
1  #include <iostream>
2       머리부
3  int main()                          몸체 블록
4  {
5    // 표준 출력 스트림으로 문장을 출력함
6    std::cout << "나의 첫 번째 C++ 프로그램"
                 << std::endl;
7    return 0;
8  }
```

• (6) 입출력 스트림 (1/2)

• std::cout 객체

- 표준 출력 스트림 객체
- 데이터를 문자열로 변환하여 출력함
- 출력 연산자(삽입 연산자) : <<

```
std::cout << "나의 첫 번째 C++ 프로그램";
```

```
int a = 10;
std::cout << "a의 값은 ";
std::cout << a << "입니다." << std::endl;
```

• (6) 입출력 스트림 (2/2)

• std::cin 객체

- 표준 입력 스트림 객체
- 문자열을 입력 변수의 자료형의 값으로 변환하여 입력함
- 입력 연산자(추출 연산자) : >>

```
int a;
char str[100];
std::cin >> a >> str;
```

• (7) 명칭공간 (1/6)

• 명칭공간이란?

- 특정한 이름들이 인식되는 프로그램의 부분

명칭공간의 정의

명칭공간 이름

```
namespace myNSpc {
   int count; // 명칭을 선언하는 문장 나열
}
```

명칭공간에 정의된 명칭의 사용

```
myNSpc::count = 0;
```

• (7) 명칭공간 (2/6)

• 명칭공간이란?

- 동일한 명칭이라도 서로 다른 명칭공간에서 정의되었다면 별개의 것으로 구분함

⇨ 여러 프로그래머가 작성한 프로그램을 결합하여 완성된 프로그램을 만들 경우 각자 필요한 명칭을 독립적으로 만들어 사용할 수 있음

- 전역 명칭공간 : 특정 명칭공간에 속하지 않는 기본 명칭공간
- `std` 명칭공간 : 표준 C++ 라이브러리의 명칭들이 정의되어 있는 명칭공간

• (7) 명칭공간 (3/6)

• 명칭공간 사용 예

```
1  #include <iostream>
2
3  namespace myNSp1 { int n = 10; }
4  namespace myNSp2 { int n = 20; }
5  int n = 30;
6
7  int main( )
8  {
9      int n = 40;
10     std::cout << myNSp1::n << std::endl;
11     std::cout << myNSp2::n << std::endl;
12     std::cout << ::n << std::endl;
13     std::cout << n << std::endl;
14     return 0;
15 }
```

• (7) 명칭공간 (4/6)

• 'using'을 이용한 명칭공간 사용

- 특정 명칭공간이나 명칭공간 내의 특정 이름을 자주 사용하는 경우 명칭공간 지정을 간소화 할 수 있음

예

```
using namespace std;
```

예

```
using std::cout;
using std::endl;
```

• (7) 명칭공간 (5/6)

• 명칭공간 사용 예

```
1  #include <iostream>
2  using namespace std;
3  namespace myNSp1 { int n = 10; }
4  namespace myNSp2 { int n = 20; }
5  int n = 30;
6
7  int main( )
8  {
9      int n = 40;
10     cout << myNSp1::n << endl;
11     cout << myNSp2::n << endl;
12     cout << ::n << endl;
13     cout << n << endl;
14     return 0;
15 }
```

• (7) 명칭공간 (6/6)

• 명칭공간 사용 예

- FirstStep.cpp

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5    // 표준 출력 스트림으로 문장을 출력함
6    cout << "나의 첫 번째 C++ 프로그램"
         << endl;
7    return 0;
8  }
```

• 제2강

•C++ 언어의 기초 (1)