CHAPTER 01. C++ 첫걸음

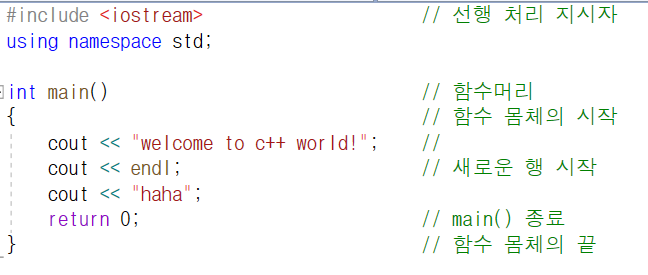
## C언어

* 절차지향적 언어이다.
* 하드웨어를 효율적으로 제어할 수 있는 간결한 언어
* 절차지향적 언어는 프로그래밍에서 데이터보다 알고리즘에 더 치중한다는 뜻
* 절차지향적 언어는 모듈 구성이 용이하며 구조적인 프로그래밍이 가능하다.
* 컴퓨터의 처리구조와 유사해 실행속도가 빠르다.
* 유지보수가 어렵고 대형 프로젝트에 부적합하다.
* 코드가 길어지면 가독성이 무척 떨어지며 이해하기 힘들다.

## C++

* 객체지향적 언어이다.
  + 객체: 추상화의 산물.
  + 추상화: 변수나 메소드들을 캡슐화하고 그것을 외부에서 사용 못하게 information hiding하는 과정을 말한다.
* 객체 지향 프로그래밍은 데이터를 강조하므로 언어 자체를 해결해야 할 문제에 맞춘다. 즉, 해결해야 할 문제의 특성에 맞게 데이터형 자체를 설계한다.

CHAPTER 02. C++ 시작하기



## main() 함수

### 인터페이스 역할을 하는 함수 머리

: 함수 머리는 호출 함수와 피호출 함수의 인터페이스를 나타낸다. 함수 이름의 앞 부분을 함수 리턴형(return type)이라고 한다. 이것은 피호출함수가 호출 함수로 다시 넘겨주는 정보의 흐름을 나타낸다. 함수 이름 뒤에 있는 괄호 안의 부분을 인자 리스트(argument list) 또는 매개변수 리스트(parameter list)라고 한다.

이것은 호출 함수가 피호출함수로 넘겨주는 정보의 흐름을 나타낸다.

: 다른 함수에 의해 호출된 함수는 자신을 호출한 함수에게 값을 리턴할 수 있다. 이 값을 리턴값(return type)이라고 한다.

: main 함수의 함수머리 ‘int main()’은 main()과 OS 사이의 인터페이스를 나타낸다.

: int main(void) 라고 해도 되고 void main(void)라고 해도 된다. 그러나 void 리턴형은 일부 시스템에서는 동작하지 않는다. C++ 표준이 아니기 때문이다.

: 모든 C++ 프로그램에는 메인함수가 반드시 하나 있어야 한다. 그러나 DLL모듈을 작성할 때와 같이 독립된 프로그램이 아니면 메인 함수가 필요 없다. 로봇의 컨트롤러 칩과 같은 특수한 황경에서도 필요하지 않을 수도 있다. 그러나 일반적인 독립형 프로그램에는 반드시 메인함수가 필요하다.

## C++ 전처리기와 iostrem 파일

: C와 마찬가지로 C++도 전처리기(preprocessor)를 사용한다.

: 전처리기는 컴파일을 하기 전에 소스 파일에 대해 미리 어떤 처리를 수행하는 프로그램이다.

: #include 전처리 지시자를 보통 사용한다. 이 지시자는 전처리기에게 특정 파일의 내용을 프로그램에 추가하라고 지시한다. 이와 같이 컴파일되기 전에 소스 코드에 텍스트를 추가하거나 텍스트를 대체하는 것이 전처리기가 수행하는 기본적인 역할이다.

: C++ 헤더파일에는 확장자를 사용하지 않기로 했다. 그리고 c 헤더파일의 이름에서 .h를 뺀 것이다. 예를 들어 math.h 헤더파일을 C++에서는 츰소로 이름이 바뀌었다.

## namespace