

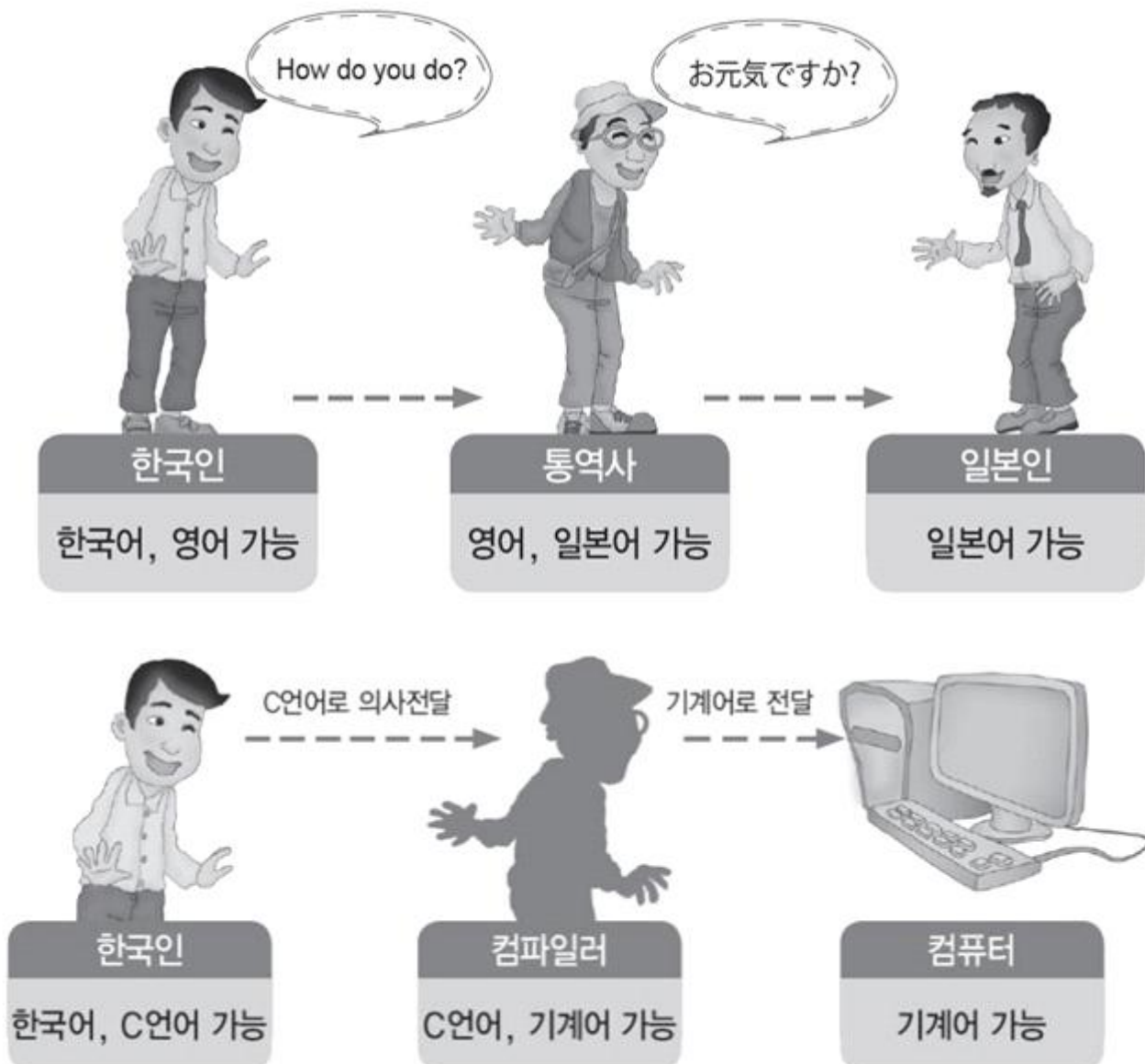
1주차 1차시 이것이 C언어다.

【학습목표】

1. C언어의 개론적 이야기에 대해 설명할 수 있다.
2. C프로그램의 완성과정에 대해 나열하여 설명할 수 있다.

학습내용1 : C언어의 개론적 이야기

1. C언어는 프로그래밍 언어이다.



* 프로그래밍 언어란 무엇인가?

- 사람과 컴파일러가 이해할 수 있는 약속된 형태의 언어를 의미 한다. C언어도 프로그래밍 언어 중 하나이다.

* 컴파일러는?

- 프로그래밍 언어로 작성된 프로그램을 컴퓨터가 이해할 수 있는 기계어로 번역하는 역할을 한다.

* 기계어(Machine Language)란?

- 컴퓨터가 이해할 수 있는 0과 1로 구성된 형태의 언어를 뜻함.

* C언어를 공부한다는 것은?

- 문법을 이해하는 것. 표현능력을 향상시키는 것. 많이 사용할수록 표현에 능숙해진다. 다른 이의 표현을 참조할수록 표현이 부드러워진다.

2. C언어의 역사와 특징

* C언어의 역사

- 1971년경 UNIX라는 운영체제의 개발을 위해 Dennis Ritchie와 Ken Thompson이 함께 설계한 범용적인 고급 (high-level)언어.
- 근원: ALGOL 60(1960) ▶ CPL(1963) ▶ BCPL(1969) ▶ B언어(1970)

* C언어 등장 이전의 유닉스 개발

- 어셈블리(assembly) 언어라는 저급(low-level)언어로 만들어졌었다.

그런데 어셈블리 언어는 하드웨어에 따라서 그 구성이 달라지기 때문에 CPU 별로 유닉스를 각각 개발해야만 했다.

* C언어 등장 이후 유닉스 개발

- C언어의 구성은 CPU에 따라 나뉘지 않기 때문에 CPU별로 유닉스를 각각 개발할 필요가 없다.

* 고급언어? 저급언어?

- 사람이 이해하기 쉬운 언어는 고급언어, 기계어에 가까울 수록 저급언어.
- C언어는 고급언어이면서 메모리에 직접 접근이 가능하기 때문에 저급언어의 특성도 함께 지닌다고 이야기 한다.

3. C 언어의 장점

* C언어는 절차지향적 특성을 지닌다. 따라서 쉽게 익숙해질 수 있다.

- 인간의 사고하는 방식과 유사하다.

* C언어로 작성된 프로그램은 이식성이 좋다.

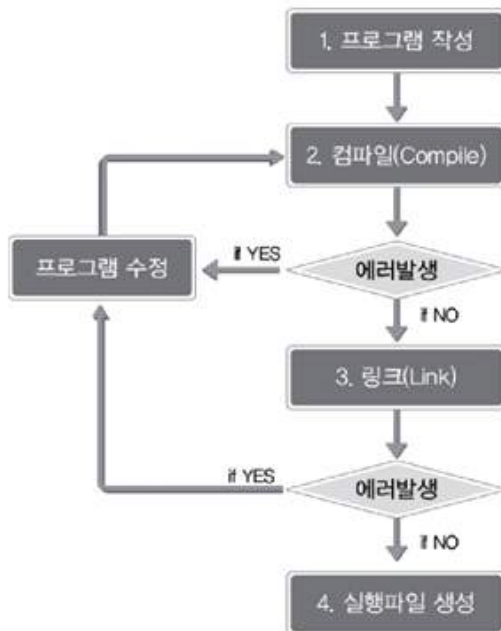
- CPU에 따라 프로그램을 재작성할 필요가 없다. 그러나 근래에는 C언어보다 이식성이 훨씬 뛰어난 언어들이 등장하고 있어서 장점으로 부각시키기에는 한계가 있다.

* C언어로 구현된 프로그램은 좋은 성능을 보인다.

- C언어를 이용하면 메모리의 사용량을 줄일 수 있고, 속도를 저하시키는 요소들을 최소화 할 수 있다.
- 단, 잘못 구현하면 오히려 성능이 좋지 못한 프로그램이 만들어지기도 한다.

학습내용2 : 프로그램의 완성과정

1. C 프로그램 완성과정의 전체적인 이해



- 첫 번째 단계 : 프로그램의 작성
- 두 번째 단계 : 작성한 프로그램의 컴파일
- 세 번째 단계 : 컴파일 된 결과물의 링크

2. VC++에서의 솔루션 생성 (1)



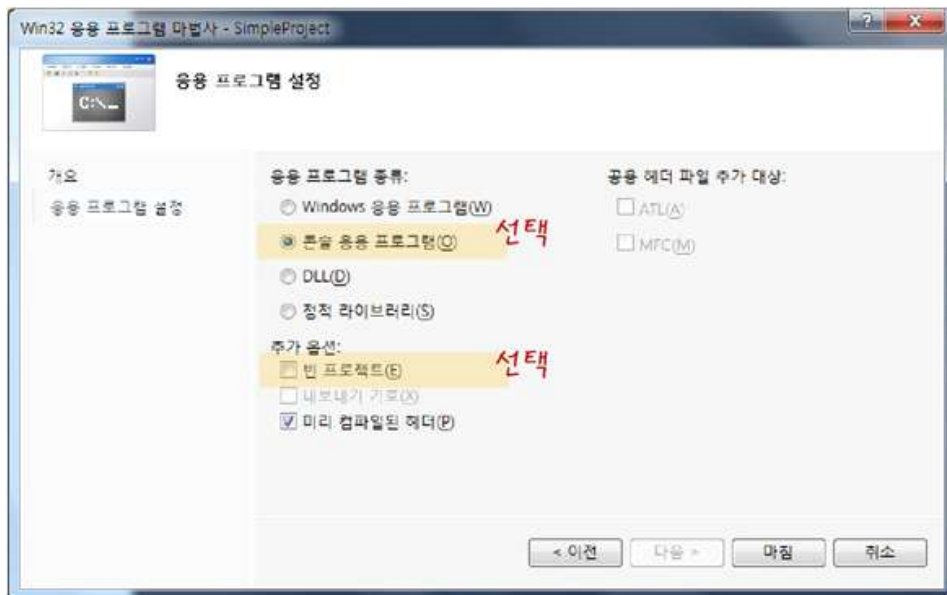
Visual C++ Express Edition의 실행

3. VC++에서의 솔루션 생성 (2)



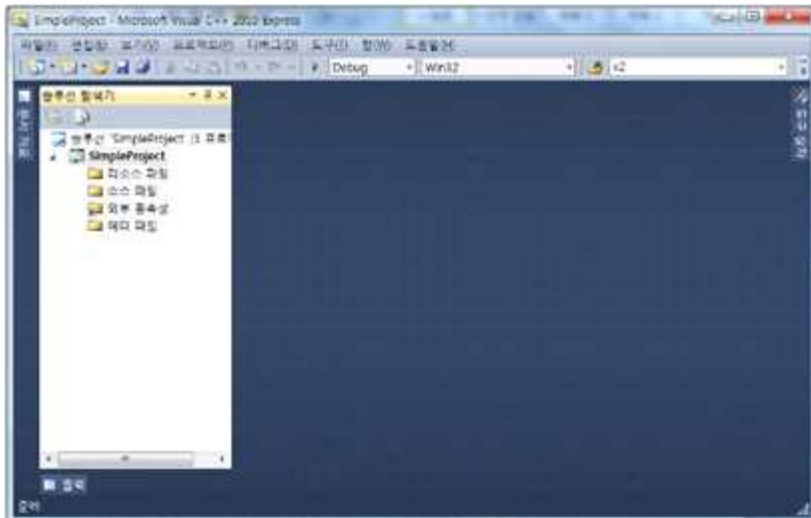
- 프로그래밍을 위한 작업공간의 마련 파일(F) → 새로 만들기(N) → 프로젝트(P)
- 프로젝트의 이름을 SimpleProject로... 프로젝트의 이름을 입력하면 솔루션의 이름은 자동으로 입력된다.

4. VC++에서의 솔루션 생성 (3)



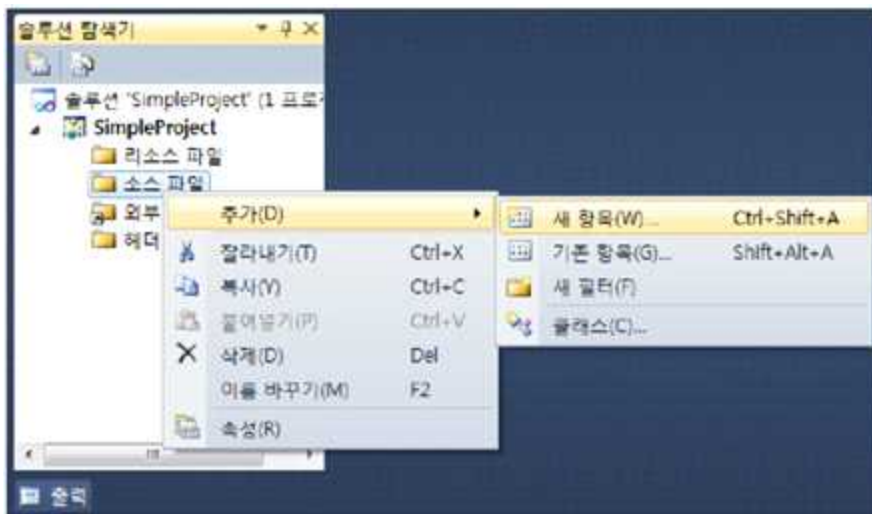
- 작업공간의 마련을 위한 최종 선택 : 응용 프로그램의 종류에서 콘솔 응용 프로그램, 추가 옵션에서 빈 프로젝트 선택

5. VC++에서의 솔루션 생성 (4)



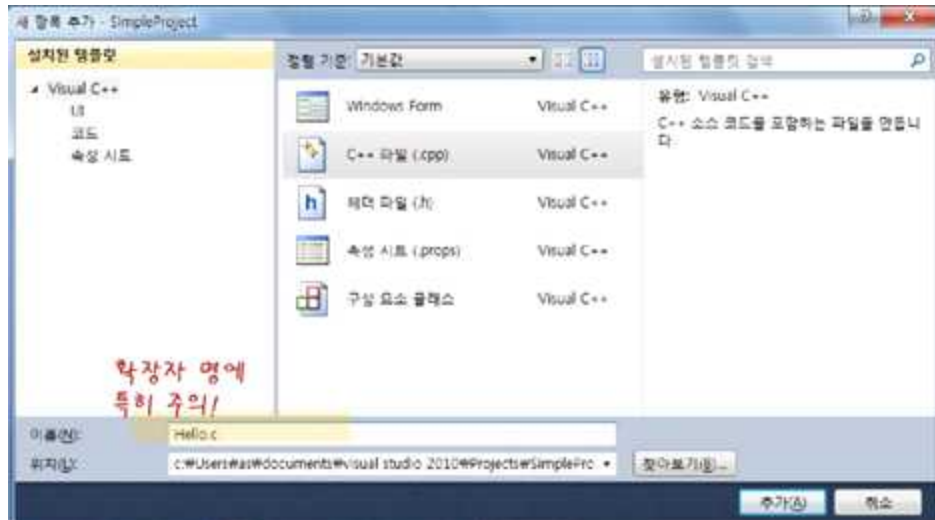
- 작업공간의 마련 완료
- 솔루션 탐색기에서 솔루션과 프로젝트가 생성되었음을 확인할 수 있다.

6. 소스파일의 생성 (1)



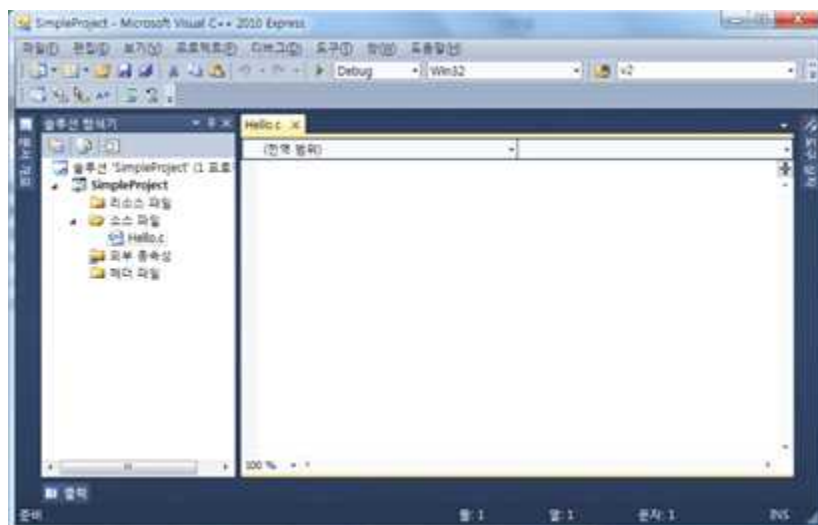
- 새로운 소스파일을 생성하여 프로젝트에 추가
- 솔루션 탐색기의 소스 파일 위치에서 마우스 오른쪽 버튼을 눌러서 추가

7. 소스파일의 생성 (2)



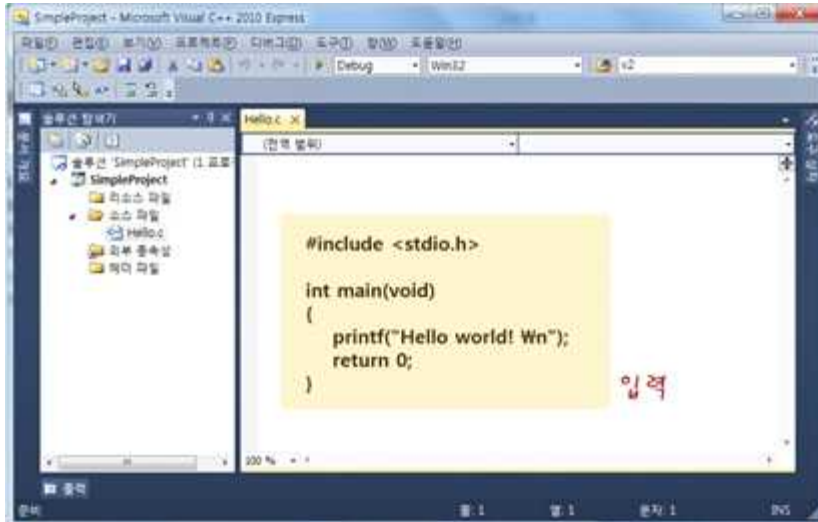
- 확장자가 .c인 소스파일 생성
- 확장자에 따라서 컴파일의 형식을 결정! 확장자를 .c로 해야 C언어 기반으로 컴파일 한다.

8. 생성된 소스파일



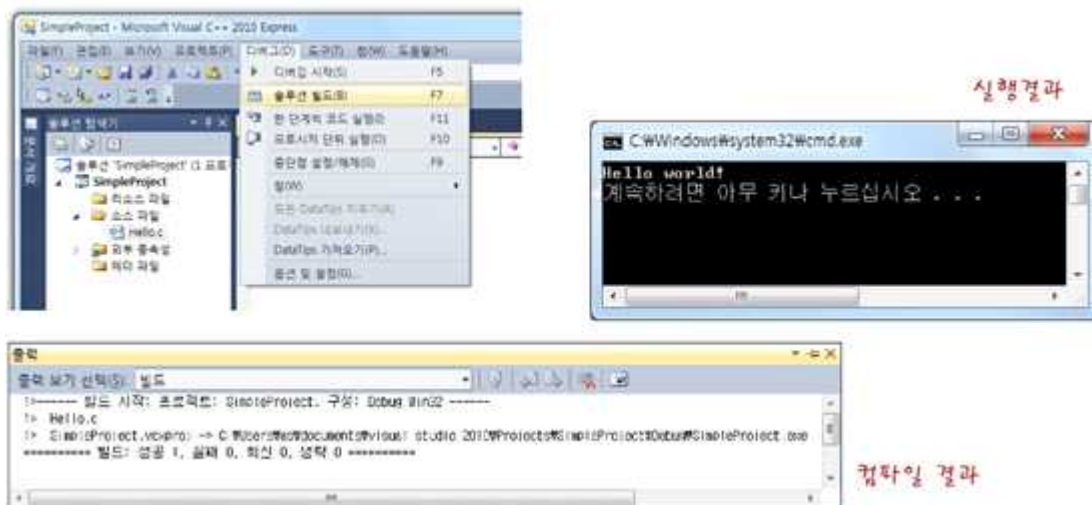
- 소스파일 생성 완료

9. 생성된 소스파일과 프로그램의 입력



- 소스코드의 입력
- 생성된 소스파일에 소스코드를 입력한다

10. 컴파일 및 실행결과 확인



- 컴파일과 실행결과의 확인
- 생성된 소스파일에 소스코드를 입력한다.

학습내용3 : Hello world 들여다 보기 (1)

① C언어의 기본단위인 '함수'의 이해

② C언어의 기본단위는 함수이다. : 함수를 만들고, 만들어진 함수의 실행순서를 결정하는 것이 C언어로 프로그램을 작성하는 것이다.

③ 함수의 기본특성

- 수학적으로 함수에는 입력과 출력이 존재한다.

④ C언어의 함수

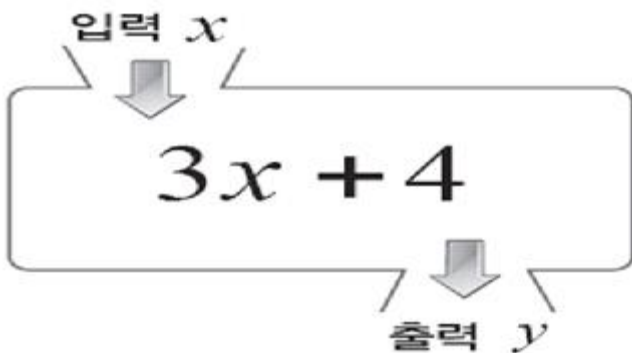
- C언어의 함수에도 입력과 출력이 존재한다.

⑤ C언어의 함수와 관련된 용어의 정리

- 함수의 정의 만들어진 함수, 실행이 가능한 함수를 일컬음

- 함수의 호출 함수의 실행을 명령하는 행위

- 인자의 전달 함수의 실행을 명령할 때 전달하는 입력 값



【학습정리】

1. 프로그래밍 언어란 사람과 컴파일러가 이해할 수 있는 약속된 형태의 언어를 의미 한다.
2. C언어는 절차지향적 특성을 지닌다. 따라서 쉽게 익숙해질 수 있다,
3. C언어로 작성된 프로그램은 이식성이 좋다.
4. C언어로 구현된 프로그램은 좋은 성능을 보인다.