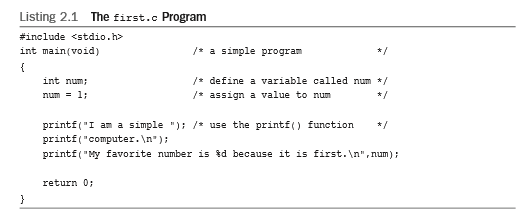
2 C 소개

2.1 C의 간단한 예제

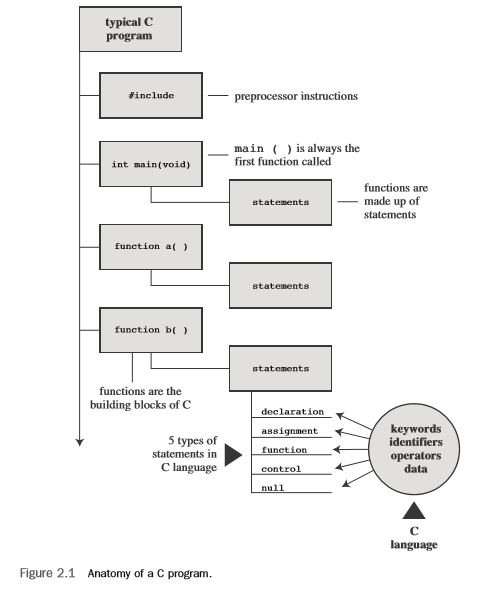
□○•★▶



□ program에서 \n 과 %d 는 어떤 용도로 사용하나?

* Listing 2.1를 실행하면 화면이 깜박거림(flash)과 화면이 사라지는데 어떻게 해결하나?
* Return 0; 앞에 getchar(); 또는 system(“pause”); 입력하여 실행 결과가 화면에 나오도록 만드는 실습

2.2 예제 설명



2.2.1 Pass 1: 간략한 개요

□#include <stdio.h> 또다른 file을 포함(include)한다

* compiler에게 stdio.h라는 file에 있는 object code를 include하도록 지시/stdio.h는 C compiler packages의 표준 입출력을 정의하는 code를 포함함

□int main(void) <- function 이름

* C programs은 1개 이상의 functions으로 구성/ function을 C program의 기본적인 module이라고 함
* main이라 불리는 function으로 구성됨
* int 은 main() function가 정수를 return한다는 의미
* void 은 function main()이 어떤 arguments도 전달하지 않는다는 뜻

□/\* a simple program \*/ <- comment

* 기호 /\* 와 \*/은 comments와 code를 구분하는 경계

□{ <- function의 body 의 시작임을 표기하는 것

* 열린 brace는 function을 구성하는 statements들의 시작임을 알리는 것
* 닫힌 brace ( } )는 function definition의 끝을 표시하는 것

□int num; <- declaration statement이라고 함

* num라고 하는 variable 을 사용하는 것을 선언
* num 변수는 int (integer) type이라고 함

□num = 1; <- assignment statement이라고 함(치환문)

* num라고 부르는 variable에 1을 assign하는 것

□printf("I am a simple "); <- function call statement이라고 함/함수 호출 문장

* “I am a simple” 을 screen에 display함
* printf()는 standard C library의 일부임
* function이라는 용어를 사용
* program에서 function을 사용하는 것을 calling a function(함수 호출)이라 함

□printf("computer.\n"); <- 또다른 function call statement

* \n 는 화면에 다음 줄(new line)에서 display를 하라는 뜻임—즉 cursor가 화면의 다음 줄로 넘어가라는 뜻

□printf("My favorite number is %d because it is first.\n", num);

* %d 는 variable num의 값을 정수 형태로 출력하라는 것

□return 0; <- return statement

* C function는 main() function을 호출한 다른 program에게 0 값을 return함

□} <- function의 끝

○ program이 닫힌 brace로 끝이라는 표기임

2.2.2 Pass 2: Program의 상세 내용

★ #include Directives 와 Header Files

□#include <stdio.h>

○ C preprocessor directive라고 함

* C compiler는 source code compiling을 하기 전에 사전 예비적인 작업을 하는 데 이것을 preprocessing이라고 함
* stdio.h file은 입출력 function의 code를 포함하고 있음, printf()는 화면에 display하기 위한 code를 포함

○ standard input/output header 라고 부름

★ The main() Function

▶ int main(void)

□ C program은 항상 main()이라는 불리는 function으로 시작함

○ main()부터 source code가 작성되어야 함

* functions이 program의 기본적인 module이라는 것을 기억해야 함

○ int 은 main() function의 return type임을 나타내는 것

○ return은 어디로? Main() 함수를 호출한 operating system에 값을 반화

* main()의 괄호는 main function에 전달되는 parameter(매개변수)를 말함

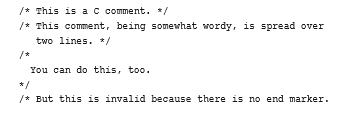
○main() // 옛날 C coding 방식

○C99와 C11 standards는 main() {}를 지원하지 않음

★ Comments

▶ /\* a simple program \*/

□ comments를 사용하는 것은 다른 source code 개발자에게 당신이 작성한 내용을 쉽게 파악할 수 있게 해준다





**★ Braces, Bodies, Blocks이라는 용어들**

▶ { ... }

□ braces 은 main() function을 delimit(범위를 정한다)

□ 모든 C functions는 function body의 시작과 끝을 표기하기 위해 brace를 사용한다

* Braces는 function 내에 어떤 statements를 하나의 단위인 block으로 묶어 주기 위해서도 사용됨

★ Declarations

▶ int num;

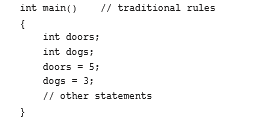
□ 변수 선언 문장을 declaration statement이라고 부름

○ num라는 변수(variable)를 사용

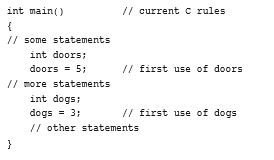
○ int는 num를 integer라는 type를 갖는다는 것을 선언하는 것

○ int 가 data type의 예제임

* compiler는 num라는 variable를 위한 메모리 상의 적합한 storage space를 할당함 .
* semicolon 은 하나의 C statement 의 끝을 나타내는 표기임
* semicolon은 statement의 일부임, 단순히 statements의 분리자로 사용된 것이 아님
* int는 기본적인 C data types의 하나를 보여주는 C keyword
* Keywords는 C 언어를 표현하는 데 사용되는 단어로서 programmer가 다른 용도로 사용할 수 없음.
* 단어 num는 identifier라고 함 – 즉 variable을 사용하기 위해 사용된 name임
* **declaration**은 특정 **identifier가**  computer memory의 특정 **location**를 가리키도록 하고 이 **location**에 저장된 **data type**을 명시하는 것
* C에서 모든 variables은 사용되기 전에 declared되어야 함
* 일반적으로 C는 program 이나 block의 시작 부분에 variables이 declared되도록 함



○ C99 and C11



* 처음 C 언어를 공부할 때 알아야 하는 사항: 1) data types이 무엇인가? 2) variable의 name을 선택할 때 어떤 choices가 있는가? 즉, 변수 이름을 사용할 때 지켜야 하는 규칙이 있는가? 3) 왜 variables들을 모두 선언해야 하는가?

★ Data Types

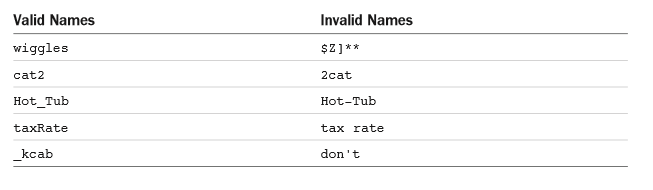
□ C 언어는 여러 종류의 data type을 지원함: integers, characters, and floating point

* variable를 integer로 선언하는 것은 computer가 정수 값을 store, fetch, and interpret하는 것을 말함

★ Name Choice

□ 변수의 첫번째 문자는 항상 letter 또는 underscore로 시작해야 함.

□ variables에 대한 names(identifiers)로서 올바른 예제



* C names은 case sensitive함, 대소문자를 구분한다는뜻으로 stars는 Stars와 STARS와 다른 이름임 .

★ Variables를 declare해야 하는 좋은 이유

□ 모든 variables를 한 곳에 모아둠으로서 source code의 가독성을 더 쉽게 함

* 변수 name으로 충분히 의미가 전달되지 않으면 variable이 표현하는 것을 comments로 기술함. 이런식으로 comments를 기술하는 것을 program의 Documenting이라 하며 좋은 programming 기법중의 하나임 – documenting도 programming technique이라고 보는 것임

□ program이 시작할 때 어떤 data를 사용하는가를 알려줌

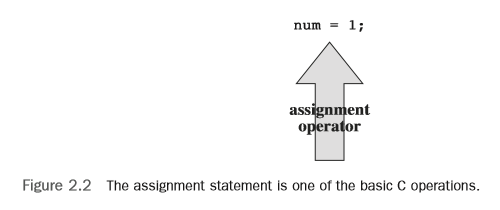
★ Assignment

□num = 1;

○ assignment statement라고 함

○ “variable num에 1을 assign함 .”

* 나중에 num에 다른 값을 assign할 수 있음, 이런 이유로 num를 variable라고 부르는 것임

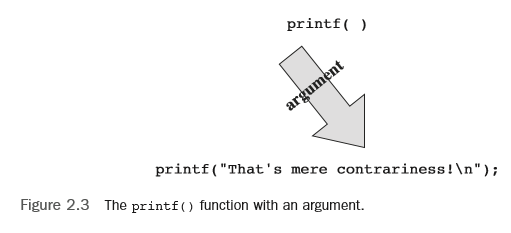


★ The printf() Function



□ printf() 라는 standard C function 을 사용하는 것임 . 괄호는 printf가 function name이라는 것을 나타냄

* 첫번째 printf()는 “I am a simple “을 passes함. 전달되는 문자열을 argument라고 함 (C는 actual argument 와 formal argument 용어를 사용: actual argument는 printf()로 전달된 값을 나타냄, formal argument는 호출된 함수에서 전달된 값을 나타내는 변수임)



□ printf()는 C에서 function을 어떻게 call 또는 invoke 하는지를 보여주는 예제임.

○ 함수 호출이 되면 control이 호출된 function : printf()으로 넘어감

* 호출된 function이 완료되면 control이 원래 호출 함수 calling function— main()으로 넘어감

□ printf("computer.\n")

* \n 기호는 새로운 line에서 display를 시작하는 뜻임: newline character

□ newline 문자가 escape sequence의 예제임

* \t는 Tab, \b는 Backspace를 위한 escape sequence라고 함
* escape sequence 문자는 backslash character, \로 시작함

□ printf("My favorite number is %d because it is first.\n", num);

* %d는 num의 값이 print될 위치(placeholder) 와 숫자 표기법을 나타냄
* %는 display되는 문자열의 해당 위치를 나타내는 것임, d는 그 위치에 variable를 print 하는 데 십진법 decimal (base 10)으로 display하라는 것임
* %d를 print되는 variable의 format 이라고 함
* printf()에서 f는 formatting print function이라는 것을 나타내는 것임
* data의 각 type은 specifier를 가짐, %f 등을 말함

★ Return Statement

□ return 0;

* int main(void)에서 int는 main() function 이 integer를 return한다는 것을 나타냄

2.3 간단한 Program의 structure

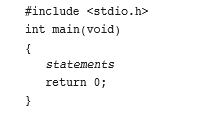
□ program 은 한 개 이상의 functions으로 구성되며 그중의 하나가 main() 임

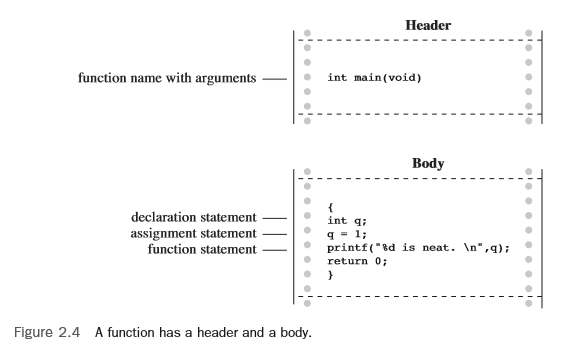
□ function 기술은 header와 body로 구성됨

□ function headers 는 function name, function에 전달되는 정보 type, function에 의하여 return되는 값에 대한 type을 기술

□ function body 는 braces ( {} )로 둘러싸이며 한개 이상의 statements 구성, 각 statement는 semicolon에 의해 끝남

* declaration statement - 사용되는 variable의 이름과 type을 선언
* assignment statement – variable에 value를 주는 것.
* printf() function를 호출
* print statements가 function call statements 예제임
* main() 함수는 return statement 으로 종료





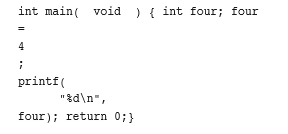
2.4 Program을 좀더 Readable하게 만들기

□ 작성된 programs이 readable하다 – 좋은 programming 습관.

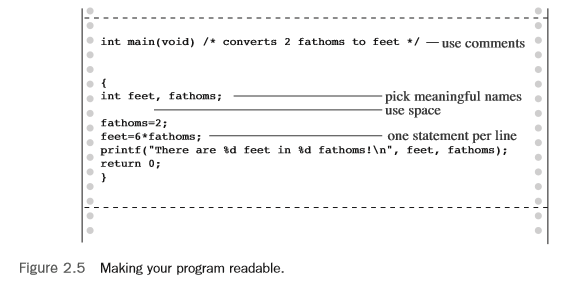
* readable program은 더 이해하기 쉽고 고치거나 correct하기가 쉽다

□ readability를 향상시키는 기법

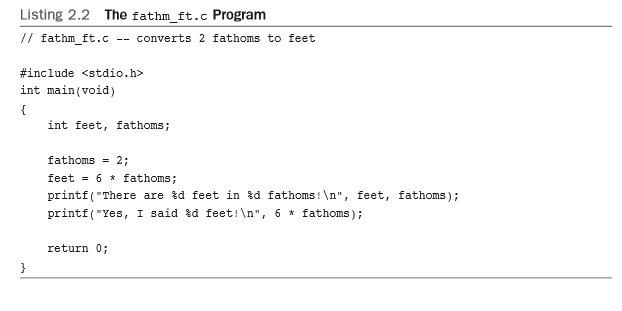
* 의미 있는 variable 이름을 사용하고 comments 기술하기
* function 사이에 빈줄 사용
* function의 선언부와 실행 statement 사이에 빈줄 사용
* 한 line에 한 statement 작성: readability 습관후
* C 는 free-form format임
* 다음 code는 오류 없으나 이해하기 어려운 code:



* program logic은 readable하게 code를 작성하면 더 이해하기가 쉽다



2.5 C program 작성시 습관화 할 부분



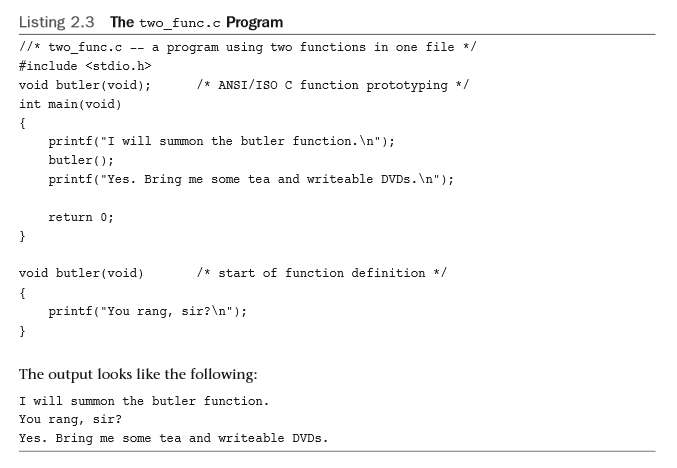
★ Documentation

□ program은 comment로 시작- filename과 program 목적을 기술

★ 한줄에 변수를 같이 선언

□ int feet, fathoms;

★ 복수개 Functions 사용시



□ function prototype: compiler에게 사용될 functions을 알려주는 것

void butler(void);

* prototype은 compiler에게 function가 사용된다고 알려주는 것: function declaration

□ function call

butler();

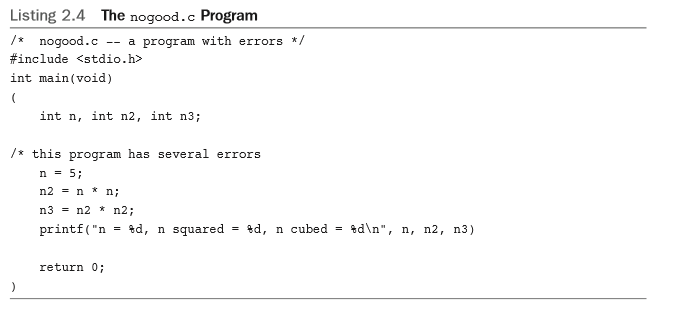
□ function definition : function의 source code

□ C 표준: function을 사용하기 전에 function prototype을 선언하는 것을 권고

* 6장: non-void function의 사용예
* 9장: function 자체를 상세히 공부

2.7 Debugging 소개

□ Program error: bug라 부름, bug를 발견하고 fix하는 것을 debugging이라 함



2.7.1 Syntax Errors/구문 오류

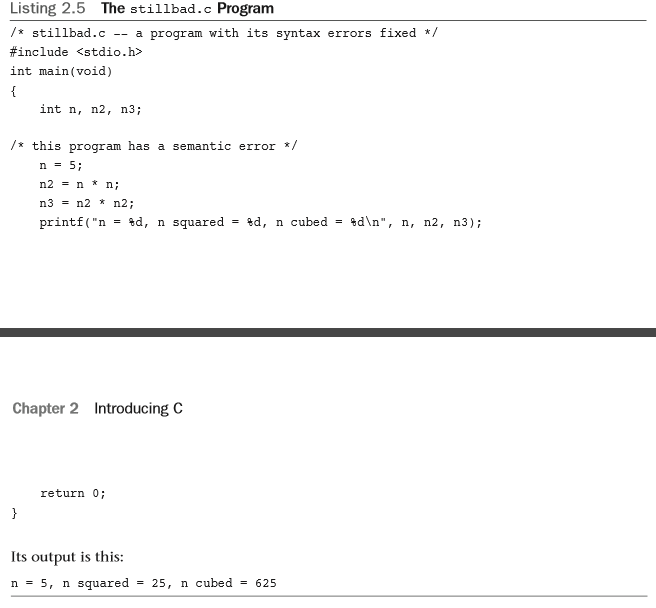
□ syntax error: C의 규칙을 따르지 않는 것

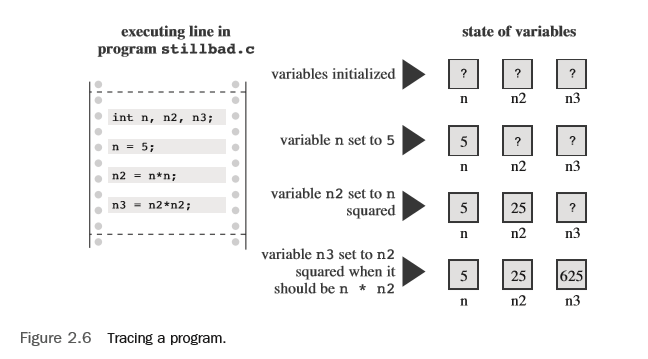
* 영어 문법 오류와 유사
* C syntax errors
* main() 함수에서 brace 대신에 parentheses를 사용
* int n, int n2, int n3; 에서 keyword int를 반복 사용 – 하나의 statement에서는 C keyword는 오직 한번만 사용
* comment를 표기하는데 \*/ 기호를 누락
* printf() statement의 semicolon을 누락

2.7.2 Semantic Errors/의미 오류 – 논리 오류

□ Semantic errors는 의미/논리 오류

* n3 = n3을 계산하기 위해 n3 = n\*n\*n;사용
* n = 5; n2 = n\*n; n3 = n2 \* n2; 구문 오류는 없으나 n3 = n4;와 같이 의미가 달라짐





• 그림 2.6: 각 statement가 실행되면 변수 값의 변화를 따라가는 것이 coding의 첫걸음

2.7.3 Program State

□ program에 사용된 각 variable의 값 상태: program state라 함

□ program state : program execution시에 실행되는 line에서의 모든 변수 값 상태

* snapshot이라 함

□ program state를 추적하는 방법: program을 line 별로/ function별로 실행

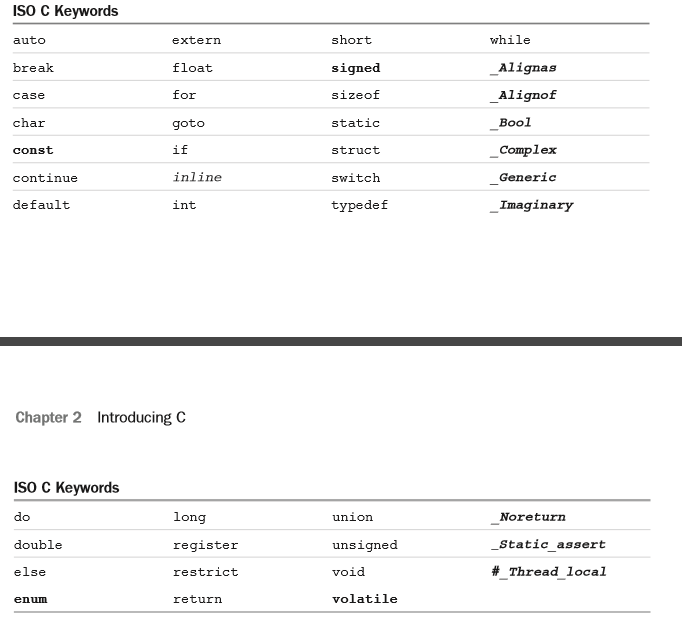
□ program의 중간 중간에 printf() statements을 추가하여 변수 값을 추적하는 방법 – 원시적 방법

□ 가장 바람직한 방법: debugger 사용

* debugger : visual studion2019 사용하여 break point 실습

★ Keywords와 Reserved Identifiers

□ Keywords



□ reserved identifiers

* Reserved identifiers는 printf()와 같은 표준 library function 이름을 말함
* 2장 학습 목표: C program의 구성 요소와 용어 이해 – socrative quiz 시행
* 2장 실습 목표: 간단한 +, -, \* 연산과 printf()를 사용하는 code의 debugging