

C 언어 기초

Jee-Hwan Ryu

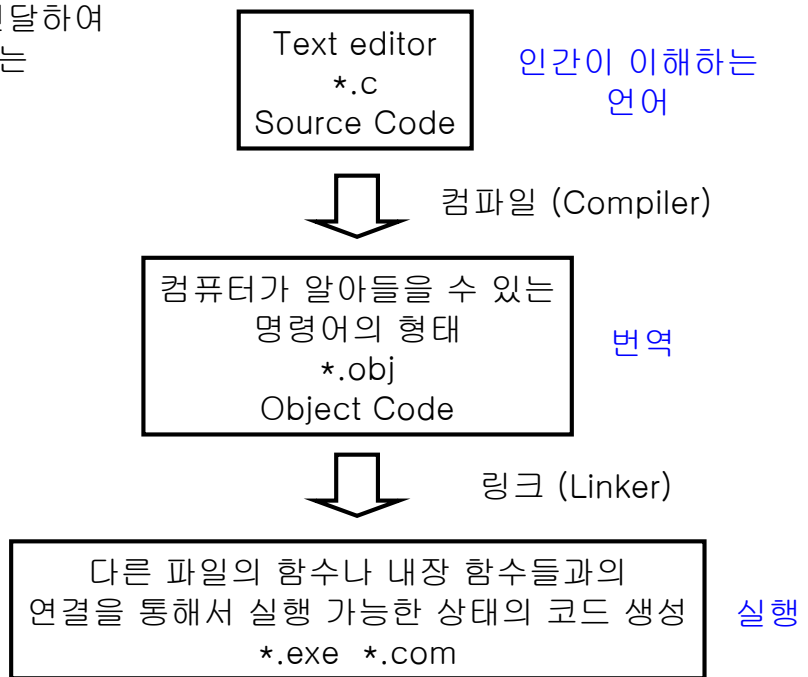
School of Mechanical Engineering
Korea University of Technology and Education

C 언어의 특징

-
- 1970년대 초 Bell 연구소에서 개발
 - 문법적 유연성 간결성
 - 고도로 구조화
 - 여러 화일로 나누어 프로그램 후 하나로 합치는 피라미드식 구조
 - 광범위한 확장성
 - 많은 기능의 루틴들을 함수형태로 만들어 라이브러리 구성
 - 하드웨어의 밀착 제어가 가능
 - 사용자가 편리하도록 만든 고급언어

프로그램을 짠다

“컴퓨터에게 인간의 의도를 전달하여
컴퓨터로 하여금 인간이 시키는
일을 하도록 만드는 것”



Korea University of Technology and Education

엄청 간단한 C 프로그램

■ 간단한 C 프로그램

```
#include <stdio.h>           ... 표준 입출력용 header 파일

main()                       ... main 함수는 반드시 포함
{                             ... 프로그램 본체는 { } 내에 작성
    printf("Programming in C is easy.\n"); ... 출력문
}
```

- 대소문자 구분
- \n : 개행문자(new line)
- 문장 끝은 semicolon(;)으로 종료
- printf(" ... ") 는 출력을 위해서 미리 작성된 프로그램으로서 library 함수라고 함.

Korea University of Technology and Education

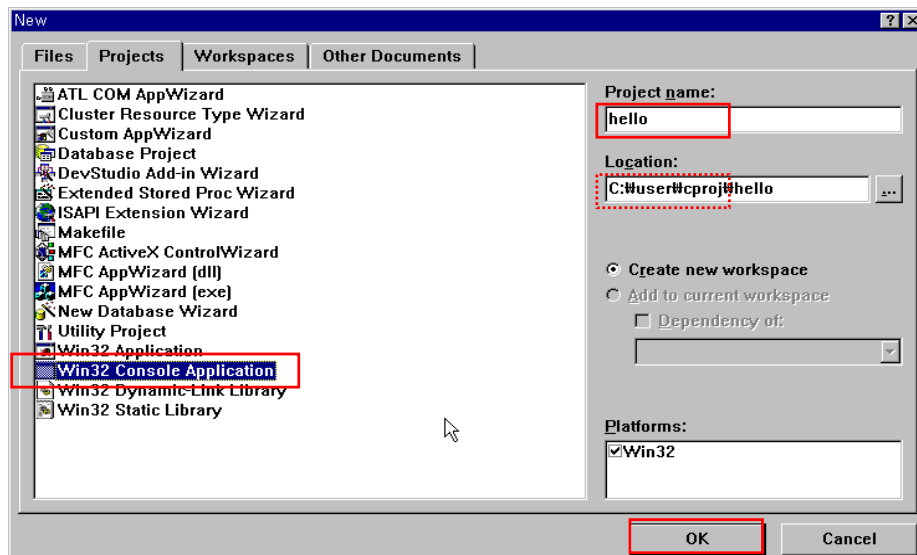
통합개발환경을 사용한 프로그램 개발

- Visual C++ 프로그램 실행
 - [시작] - [프로그램] - [Microsoft Visual Studio] - [Microsoft Visual C++]
- 새 프로젝트 파일 생성
 - [File] - [New] 메뉴 선택
 - New 대화 상자의 Projects 탭에서
 - [Win32 Console Application] 선택
 - 필요하면 Location 입력 (project 파일 생성 위치를 변경할 때)
 - Project name 입력 후 [OK]

Korea University of Technology and Education

통합개발환경을 사용한 프로그램 개발

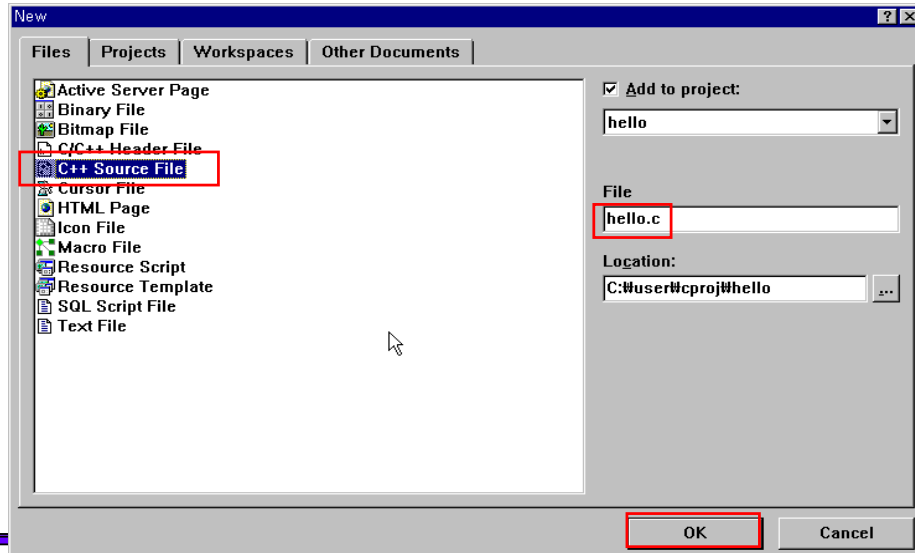
- Visual C++ 프로그램 실행
 - [시작] - [프로그램] - [Microsoft Visual Studio] - [Microsoft Visual C++]
- 새 프로젝트 파일 생성
 - [File] - [New] 메뉴 선택



Korea University of Technology and Education

통합개발환경을 사용한 프로그램 개발(2)

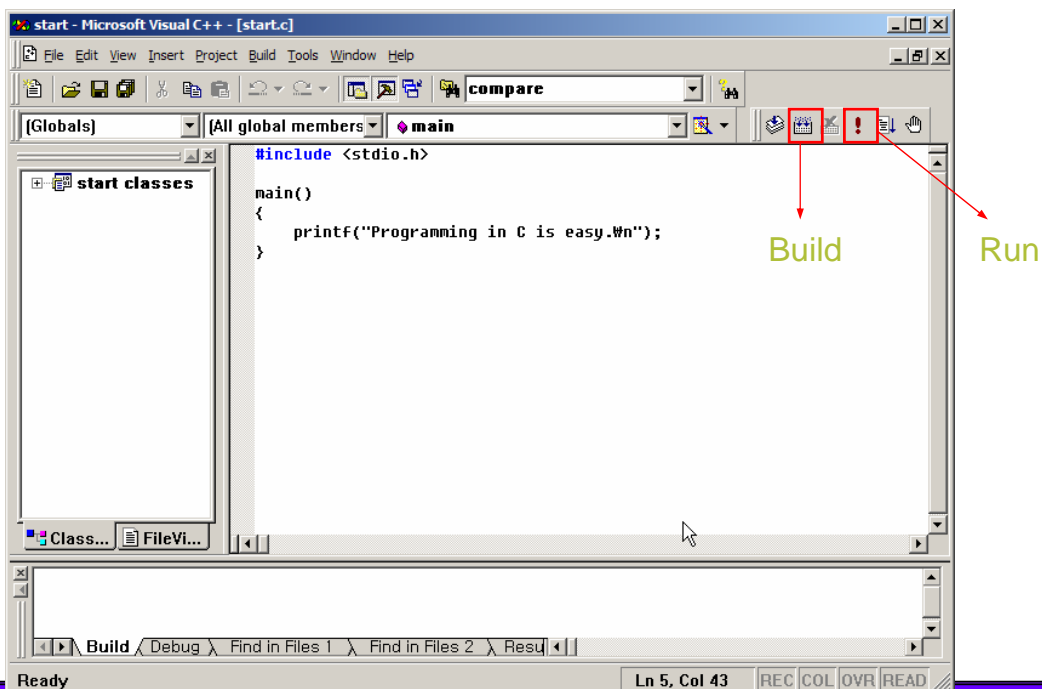
- New Project Information 대화 상자에서
 - empty project 선택 후 [Finish]
- 새 C 프로그램을 프로젝트에 추가
 - [File] – [New] 메뉴 선택



Korea University of Technology and Education

통합개발환경을 사용한 프로그램 개발(2)

- 프로그램 작성: [File] – [Save] (단축키 Ctrl+S) 메뉴 선택하여 저장

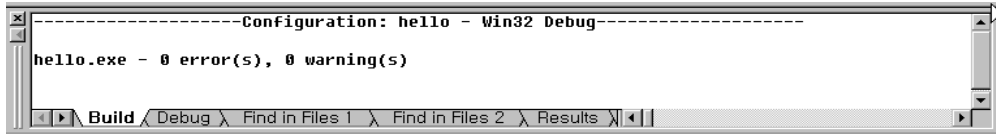


Korea University of Technology and Education

통합개발환경을 사용한 프로그램 개발(3)

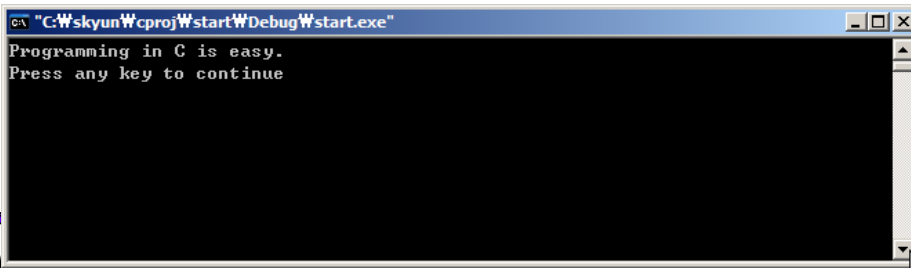
■ 컴파일 & 링크

- [Build] – [Build *.exe] (단축키 F7) 또는 [Build] – [Build All] 메뉴 선택
- 컴파일 결과가 화면 아래의 창에 출력됨



■ 실행

- [Build] – [Execute *.exe] (단축키 Ctrl+F5) 메뉴 선택



Korea University of Technology and Education

통합개발환경을 사용한 프로그램 개발(4)

■ 종료

- 변경된 내용 저장: [File] – [Save All] 메뉴 선택
- 작업 종료: [File] – [Close Workspace] 메뉴 선택

■ 프로젝트 다시 수행

- Visual C++ 실행
- [File] – [Open Workspace] 메뉴 선택
- 작업할 project에 대한 workspace 파일 선택

실습문제

- 다음과 같이 자신의 이름을 출력하는 프로그램을 작성하여 실행.

Hello, Gil-Dong Hong !!

Korea University of Technology and Education

C 프로그램의 작성과 구조이해

```
/* Sample Program */      /* 설명문(컴파일 되지 않음) */
#include <stdio.h>        /* 전처리기*/

int GetSum(int x, int y)  /* 사용자 함수 정의 */
{
    int z;
    z = x + y;

    return(z);           /* 결과값을 출력으로 되 돌려줌 */
}

void main()               /* main 함수 */
{
    int a, b, sum;        /* 변수형 선언 */
    scanf("%d%d", &a, &b); /* 입력함수 호출 */
    sum = GetSum(a,b);    /* 사용자 함수 호출 */
    printf("The sum of %d and %d is %d", a, b, sum); /* 출력함수 호출 %:위치, d: 십진수 */
}
```

위 프로그램을 바탕으로 앞으로 배우게 될 내용들을 살펴보자 !

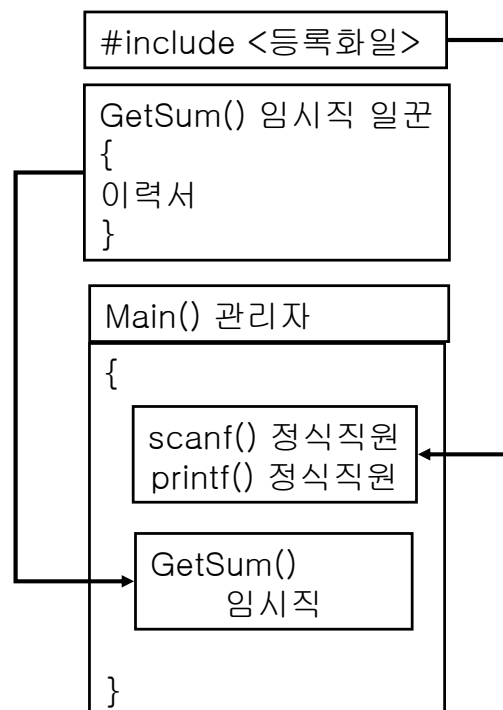
Korea University of Technology and Education

C 언어의 함수

- ‘C 언어는 **함수들의 집합**이다.’
- C 언어의 함수들은 컴퓨터에게 원하는 일들을 지시하는 명령어들의 집합으로 이루어져 **특정작업을 하는 하나의 단위**가 되며, 그 결과값을 되돌려 주기도 하는 **일종의 일꾼**이다.
- main 함수
 - 프로그램에서 반드시 있어야 하는 특수한 함수
 - 프로그램의 실행은 main 함수로부터 시작하여 main 함수에서 끝난다.
 - 여러 일꾼들을 감독하는 **관리인**
- 내장함수
 - 회사에 정식으로 고용된 일꾼과 같다 (Ex., scanf, printf)
 - 할 수 있는 일은 stdio.h 에 등록되어 있다
- 사용자 정의함수
 - 잠시 불러쓰는 임시직 일꾼
 - 할 수 있는 일은 main 함수 전에 정의 (Ex., GetSum)

Korea University of Technology and Education

함수 및 프로그램 구조



Korea University of Technology and Education

함수의 기본 형태

되돌림 자료형 함수이름 (매개변수 리스트) -> 함수의 원형

```
{  
    변수 선언 및 초기화  
    여러 C 구문들...  
}
```

- 예제의 함수들을 살펴보자

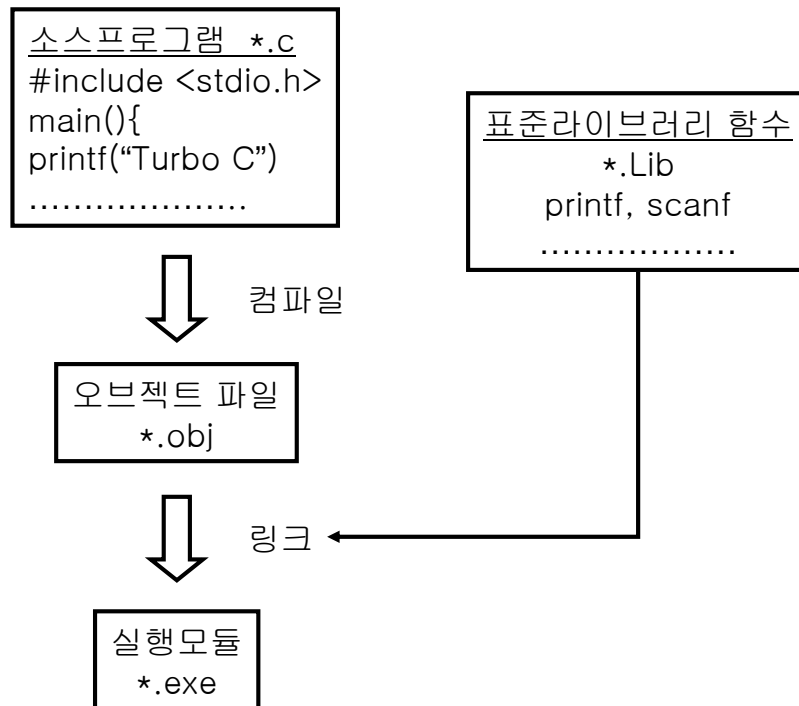
Korea University of Technology and Education

전처리기

- #include, #define 등은 전처리기 제어문
- printf() 함수나 scanf() 함수를 사용하기 위하여 이들이 등록 (선언)되어 있는 stdio.h라는 파일을 포함(include) 시키는 것
- 이를 <>로 묶어주어, C의 표준 라이브러리 헤더 파일임을 알려준다. "*.h"로 묶어주는 경우는 사용자 정의 헤더 파일
- 빈번히 쓰이는 함수들을 미리 만들어 컴파일 후 목적코드로서 라이브러리를 구성하고 있다가 링크시 실행코드에 첨가
- 이러한 함수의 원형을 나열해 놓은 헤더 파일을 프로그램의 시작부분에 둬므로서 라이브러리에 있는 기능을 사용 가능
- #define EOF -1 /* EOF 를 -1로 정의 */

Korea University of Technology and Education

프로그램의 실행 과정



식별자

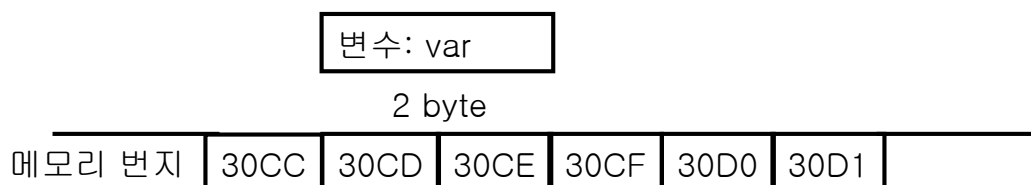
- 식별자(identifier)
 - 변수, 함수, 상수와 같은 프로그램 구성 요소의 이름
- 식별자 생성 규칙
 - 규칙: 영문자, 숫자, 밑줄문자(_)로 구성, 숫자로 시작하지 않음
 - 올바른 예: sum, x2000, tax_rate1, tax_rate2, Table, table
 - 잘못된 예: 4th, "x", tax-rate, id@host
- ANSI 표준은 31자까지 구별, 일부 컴파일러는 8자까지 구별
 - tax_rate1과 tax_rate2는 일부 컴파일러에서는 같은 식별자로 인식
- 식별자 권장사항
 - 의미를 알 수 있도록 부여하는 것이 바람직함 (예) tax
 - 두 단어를 결합한 식별자는 밑줄문자 또는 대문자로 구분함 (예) tax_rate TaxRate

예약어

- 예약어(reserved word)
 - C언어에서 특별한 용도로 미리 예약된 이름
 - 키워드(keyword)라고도 함
 - 식별자로서 사용할 수 없음
- 예약어의 예
 - 자료형: char short int long unsigned float double struct union typedef enum void const signed
 - 제어문: if else switch case default for while do break continue goto return
 - 기억장소: auto register static extern volatile
 - 연산자: sizeof
- 일부 컴파일러에서 사용하는 예약어
 - ada fortran pascal asm
 - entry near far huge

데이터 형 (자료형)

- 데이터를 처리하여 결과를 도출하는 것이 프로그램의 관심
- 이 데이터는 정수, 실수, 문자 일수도 있다. 이것이 데이터의 형(Type) 이다.
- 변수: 데이터를 프로그램 실행도중 저장해 두는 곳
- 변수는 메모리 상에서 자신의 주소를 가진다.
- int var = 10



printf() 함수를 이용한 화면 출력

```
#include <stdio.h>
void main()
{
    int a, b;
    a= 10;
    b = 5;
    printf(" multiply   divide Wn");
    printf("%8d%10dWn", a+b, a/b);
}
```

- %자리에 출력, 8,10은 출력폭에 대한 자리 수, d는 십진수, Wn 줄 바꿈
- printf(“%5.2f”, b); 실수형으로 데이터를 출력, b값을 전체 5자리(소수점 포함) 소수점 이하 2자리로 출력
- printf(“%c”, 107); ASCII코드값 107에 해당하는 ‘k’를 출력

출력함수 형식 지정자

- 정수 형식 지정자

형식지정자	출력 형식	출력 예
%d	10진수(<u>부호있는 정수</u>)	-32536
%u	10진수(<u>부호없는 정수</u>)	33000
%o	8진수	100350
%x	16진수 (소문자 a~f 사용)	12fe
%X	16진수 (대문자 A~F 사용)	12FE
<u>%ld</u> , ...	long 자료형	
<u>%hd</u> , ...	short 자료형	

- int가 16비트인 경우에 long 자료형은 %ld, %lu ...을 사용
- int가 32비트인 경우에 short 자료형은 %hd, %hu ... 을 사용함

출력함수 형식 지정자(2)

- 실수 형식 지정자

형식지정자	출력 형식	출력 예
%f	보통의 실수 표기 (지수없음)	123.456000
%e	과학용 표기 (소문자 e 사용)	1.23456e+02
%E	과학용 표기 (대문자 E 사용)	1.23456E+02
%g	%e와 %f 중 더 짧은 표기	
%G	%E와 %f 중 더 짧은 표기	
%p	포인터(주소) 값	

- 문자/문자열 형식 지정자

형식지정자	출력 형식	출력 예
%c	문자	a
%s	문자열	Korea
%%	문자 %를 출력	%

Korea University of Technology and Education

예제

- ```
int a = -200;
printf("%d %uWn", a, a);
```

 ... 음수를 %u지정자로 출력
- ```
printf("%o %xWn", 500, 500);
```

 ... 10진수를 8, 16진수로 변환
- ```
double r = 123.45678;
double s = 12340000.0;
printf("%f %e %gWn", r, r, r);
printf("%f %e %gWn", s, s, s);
```

 ... %f, %e, %g 비교
- ```
printf("%s and 100%%", "C language");
```

 ... 문자열, % 출력
- ```
char c = 'A';
printf("%c의 코드: 10진수 %d, 8진수 %o, 16진수 %x", c, c, c, c);
```

 ... 문자의 ASCII코드값 출력

*Korea University of Technology and Education*

## scanf( ) 함수를 이용한 입력

```
#include <stdio.h>
void main()
{ char a;
 printf("Enter a character : ");
 scanf("%c", &a);
 printf("%c\n", a);
}
```

```
int a, b, c, d;
d=scanf("%d%d%d", &a, &b, &c);
```

세 수를 입력시킬 때, 10 20 30과 같이 숫자 사이 한 칸씩 띄워서 구별 입력이 이루어지면 3이 d에 저장, scanf()의 리턴값은 정상적으로 입력된 항목의 수

## 입력함수 형식 지정자

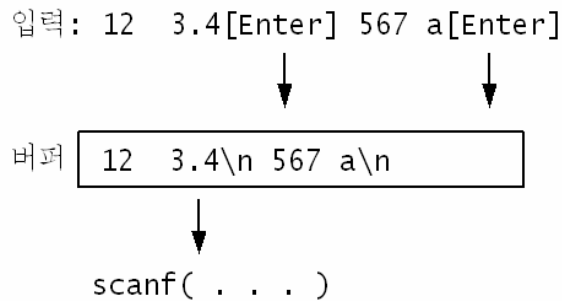
- scanf() 형식지정자

| 형식지정자    | 입력자료형      | 입력 형식     | 입력 예            |
|----------|------------|-----------|-----------------|
| %d       | 정수         | 10진수      | -123            |
| %u       | 부호없는 정수    | 10진수      | 30000           |
| %o       | 부호없는 정수    | 8진수       | 644             |
| %x       | 부호없는 정수    | 16진수      | 5f5a, 5F5A      |
| %ld, ... | long 자료형   |           |                 |
| %hd, ... | short 자료형  |           |                 |
| %f, %e   | 실수(float)  | 임의의 실수 표기 | 1.23, 3.1e-2, 5 |
| %lf, %le | 실수(double) | 임의의 실수 표기 |                 |
| %c       | 문자         |           | a               |
| %s       | 문자열        |           | Korea           |

- printf()와 비슷하지만 차이가 있음

# 라인버퍼 방식 입력

- 라인버퍼 방식 입력
  - 키보드에서 [Enter]가 입력될 때에 줄 단위로 버퍼에 저장됨
  - scanf()함수는 버퍼에서 입력 받음



# 예제

- 예제
  - 형식지정자 사이에 공백이 없어도 무방  
scanf("%d%f%f", &i, &f, &d);  
scanf("%d %f %f", &i, &f, &d);
  - (예외) 문자 입력 ... 입력: a b 인 경우에  
scanf("%c %c", &x, &y); ... x='a', y='b' 입력  
scanf("%c%c", &x, &y); ... x='a', y=' ' 입력
  - 제어문자열에 특별한 경우가 아니면 다른 문자 사용하지 않음  
scanf("%dWn", &a); (x) ... 잘못된 예  
scanf("%d : %f", &i, &f); 정수와 실수 사이에 : 입력  
(예) 5 : 12.2

## putchar( ), getchar( )

```
#include <stdio.h>
void main()
{
 putchar('a'+1);
 putchar('\Wt');
 putchar(0x44);
}
```

'a'는 문자 a의 ASCII 값 0x61, 'a'+1=0x62 로서 b의 ASCII 코드 값 0x44는 문자 D 에 해당

a=getchar(); 입력한 문자의 ASCII 코드 값을 리턴한다.

*Korea University of Technology and Education*

## C의 기본 데이터 형(type)

- 자료형(data type): 데이터의 크기를 정해주기 위한 것
  - 정수 (integer)
  - 실수 (floating point number): 부동소수점 수
  - 문자 (character)

(cf) 정수 1과 실수 1의 컴퓨터 내부 표현은 다르다.  
문자는 컴퓨터 내부에서 정수로 표현된다.

- C언어의 기본 자료형
  - int            정수
  - char          문자
  - float        실수 (보통 정밀도)
  - double      실수 (2배 정밀도)

double x;            ... x는 2배정밀도 실수 변수  
char c;              ... c는 문자 변수

*Korea University of Technology and Education*

## 정수형 (Integer)

- 정수형의 크기와 범위
  - 컴퓨터와 컴파일러에 따라서 16-bit 또는 32-bit로 표현됨.
  - 16-bit 정수:  $-32,768 (2^{15}) \sim 32,767(2^{15}-1)$
  - 32-bit 정수:  $-2,147,483,648(2^{31}) \sim 2,147,483,647(2^{31}-1)$

## 실수형 (float)

- 4 byte
  - 소수점을 포함한 숫자
  - 모두 부호를 가지고 있음
  - 유효 숫자가 6개 이하일 때 float형을 사용
  - Double이나 long double은 큰 수이거나 높은 정밀도를 요구하는 제한적인 경우에 사용
- |                       | 유효숫자 | 지수범위       | 유효숫자 x 10 <sup>지수</sup> |
|-----------------------|------|------------|-------------------------|
| ● float (32-bit 실수):  | 6자리  | -38 ~ 38   |                         |
| ● double (64-bit 실수): | 15자리 | -308 ~ 308 |                         |
- 정수형 만으로도 가능한 프로그램을 일부러 실수형을 쓰지 않는 것이 좋다.

## 문자형 (Character type)

- 가장 작게 메모리를 차지
- 작은 범위의 숫자를 다룰 때나 메모리 입출력을 할 때 쓰인다.
- 주로 ASCII문자를 표현하기 위해서 쓰임

| 정수형           | 바이트 | 범위         |
|---------------|-----|------------|
| (signed) char | 1   | -128 ~ 127 |
| unsigned char | 1   | 0 ~ 255    |

- 문자는 8-bit로 표기 (cf) 한글 한 글자는 16-bit로 표기
- char 자료형은 8-bit 정수 표기에 사용 가능
- 8-bit 정수: -128 ~ 127

## 자료형 수정자 (Modifier)

- 자료형 앞에 붙여서 자료형의 표현 범위를 변경

(예) long, short, unsigned

|                    |                            |
|--------------------|----------------------------|
| long int           | 긴 정수(32-bit)               |
| short int          | 짧은 정수(16-bit)              |
| unsigned int       | 부호 없는 정수 → 양수의 표현 범위 2배 증가 |
| unsigned long int  | 부호 없는 긴 정수(32-bit)         |
| unsigned short int | 부호 없는 짧은 정수(16-bit)        |

- 자료형 수정자와 함께 사용하는 int는 생략 가능

```
long int a;
long a; } 두 변수 선언은 같은 의미임
```

- unsigned 정수의 표현 범위

- 16-bit unsigned: 0 ~ 65,535 ( $2^{16}-1$ )
- 32-bit unsigned: 0 ~ 4,294,967,295 ( $2^{32}-1$ )

## 실수를 정수변수에 대입

---

```
#include<stdio.h>
#include<math.h> /* sqrt 정의되어 있음 */

void main()
{
 int i_result;
 float f_result;

 i_result = sqrt(2);
 f_result = sqrt(2);

 printf("The result 1 is %5dWn", i_result);
 printf("The result 2 is %5.3fWn", f_result);
}
```

---

*Korea University of Technology and Education*

## 상수 (Constant)

---

- 변하지 않는 명시적인 자료
- 정수상수: 12 056 0x4f
- 실수상수: 3.14 1.0 1.5e5
- 문자상수: 'A' 'a' '1' 'Wn' '@' '한'
  - 'a' 는 십진수 97을 의미함
  - 컴퓨터는 문자 자체가 아닌 숫자로 문자를 이해한다.
- 문자열상수: "Korea" "대한민국" "한"

---

*Korea University of Technology and Education*

## 정수 상수

- 정수 상수
  - 10진수: 1~9로 시작하여 표현 (예) 789
  - 8진수: 0으로 시작하여 표현 (예) 064
  - 16진수: 0x 또는 0X로 시작하여 표현 (예) 0x3af
- 정수 상수는 기본적으로 int 형으로 표현됨
- 자료형 지정 접미사 U, L
  - 1234 ... int형 정수
  - 1234L ... long형 정수
  - 40000U ... unsigned형 정수
  - 40000UL ... unsigned long형 정수

## 실수 상수

- 실수 상수
  - 소수점을 포함한 숫자
    - 3.14
    - 1.0 1. ... 소수부분=0 이면 소수점이하 생략 가능
    - 0.12 .12 ... 정수부분=0 이면 정수부분 생략 가능
  - 과학용 표기법
    - 6.2e3 (=  $6.2 \times 10^3$ ) 6.2E3
    - 4e-2 (=  $4 \times 10^{-2}$ ) ... 소수점이 없어도 실수 상수
- 실수 상수는 기본적으로 double형으로 표현됨
- float형 지정 접미사 F
  - 6.2 ... double형 실수
  - 6.2F ... float형 실수

# 문자 상수

- 문자 상수
  - 영문자, 숫자, 특수문자를 작은 따옴표(' ')로 묶어서 표시 (예) 'A' 'b' '5' '@' '&'
- ASCII코드
  - 문자에 대응되는 정수 값을 정한 미국 표준 코드
  - ASCII코드는 8-bit 정수로 표현
  - ASCII코드의 순서는 숫자와 알파벳 순서와 같음
  - '0'과 0은 다르다.
    - '0'은 정수 48과 같음

| ASCII | Char  | ASCII | Char | ASCII | Char |
|-------|-------|-------|------|-------|------|
| 32    | Space | 64    | @    | 96    | `    |
| 33    | !     | 65    | A    | 97    | a    |
| 34    | "     | 66    | B    | 98    | b    |
| 35    | #     | 67    | C    | 99    | c    |
| 36    | \$    | 68    | D    | 100   | d    |
| 37    | %     | 69    | E    | 101   | e    |
| 38    | &     | 70    | F    | 102   | f    |
| 39    | '     | 71    | G    | 103   | g    |
| 40    | (     | 72    | H    | 104   | h    |
| 41    | )     | 73    | I    | 105   | i    |
| 42    | *     | 74    | J    | 106   | j    |
| 43    | +     | 75    | K    | 107   | k    |
| 44    | ,     | 76    | L    | 108   | l    |
| 45    | -     | 77    | M    | 109   | m    |
| 46    | .     | 78    | N    | 110   | n    |
| 47    | /     | 79    | O    | 111   | o    |
| 48    | 0     | 80    | P    | 112   | p    |
| 49    | 1     | 81    | Q    | 113   | q    |
| 50    | 2     | 82    | R    | 114   | r    |
| 51    | 3     | 83    | S    | 115   | s    |
| 52    | 4     | 84    | T    | 116   | t    |
| 53    | 5     | 85    | U    | 117   | u    |
| 54    | 6     | 86    | V    | 118   | v    |
| 55    | 7     | 87    | W    | 119   | w    |
| 56    | 8     | 88    | X    | 120   | x    |
| 57    | 9     | 89    | Y    | 121   | y    |
| 58    | :     | 90    | Z    | 122   | z    |
| 59    | ;     | 91    | [    | 123   | {    |
| 60    | <     | 92    | \    | 124   |      |
| 61    | =     | 93    | ]    | 125   | }    |
| 62    | >     | 94    | ^    | 126   | ~    |
| 63    | ?     | 95    | _    | 127   | -    |

# 한 문자 출력

```
#include<stdio.h>

void main()
{
 int a;
 putchar('a');
 putchar('\n');
 a = 'a';
 printf("0x%x %c\n", 97, a);
 printf("%c\n", 'a');
}
```

- 한 문자는 작은 따옴표, 문자열은 큰 따옴표

## 확장열

| 확장열 | 이름             | 의미             |
|-----|----------------|----------------|
| Wa  | Alert          | 경고음            |
| Wf  | FormFeed       | 프린터에서 한 페이지 넘김 |
| Wn  | NewLine        | 다음 행의 처음       |
| Wt  | Horizontal tab | 다음 tab 위치로     |
| WW  | Backslash      | 역 슬래쉬 자체       |
| W'  | Single quote   | 작은 따옴표 자체      |
| W"  | Double quote   | 큰 따옴표 자체       |

- `Printf("WaWn!W"GoodW"!Wn");`
- 경고음 (뽁)
- `!"Good"!`

*Korea University of Technology and Education*

## 문자열 상수

- 문자열 상수
  - 연속된 문자들은 큰 따옴표(" ")로 묶어서 표시  
(예) "Korea"
- 문자열의 표현
  - 문자열은 컴퓨터 내부에서 null문자로 끝나는 연속적인 문자들로 표현  
(예) "Korea" → 'K' 'o' 'r' 'e' 'a' 'W0' 

|   |   |   |   |   |    |
|---|---|---|---|---|----|
| K | o | r | e | a | \0 |
|---|---|---|---|---|----|
- (예)
  - "a"는 'a', 'W0'로 구성되므로 'a'와 같지 않음
  - null 문자열 ""은 'W0'만으로 구성되는 문자열임

*Korea University of Technology and Education*

# 변수

- 변수(variable)
  - 값을 저장할 수 있는, 이름이 부여된 기억장소
- 변수의 선언
  - 변수를 사용하기 전에 반드시 변수의 자료형을 선언해야 함.  
int p;  
float a, b, c; ... 여러 개의 같은 자료형의 변수 선언
  - 변수를 선언하면 변수의 메모리 공간이 확보됨
  - 변수 선언 위치: 함수의 시작 부분 (다른 위치는 나중에 다룸)

```
main()
{
 변수 선언
 프로그램 실행문
}
```

# 변수의 초기화

- 변수는 초기값을 지정한 다음에 사용해야 함  
int a;  
...  
b = a; (X) ... a의 값이 정의되지 않았으므로 잘못
- 실행문에서 초기화  
int a;  
...  
a = 25;
- 변수의 초기화 선언  
int a = 25;  
float data, sum = 0.0, average; ... sum만 초기화(권장하지 않음)

## 주석

---

- 주석(comment)

`/* This is a comment */` ... `/* */`로 둘러싸인 부분

- 설명문으로서 컴파일러에 의해서 무시됨
- 프로그램 저작권, 수정 이력, 동작 설명 등의 문서화 용도로 사용

- 예

```
/* hello.c
 * 내용: 자료를 순서대로 나열한다.
 * ver 0.0: 03/6/10 작성자: 홍길동
 */
int sum; /* 자료들의 합 */
int max; /* 최대값 */
```

여러 줄에 걸쳐서 작성 가능

## 산술 연산자

---

- %: 나머지 (Modulus)
- ++: 피 연산자의 값을 하나 증가 (증가 연산자)
- --: 피 연산자의 값을 하나 감소 (감소 연산자)
- $17/5$  (정수/정수) = 3
- $17.0/5$  (실수/정수) = 3.4
- $17/5.0$  (정수/실수) = 3.4
- 전위형(prefix)
  - ++a; 먼저 a를 증가시키고 이 값을 이용해 다른 연산을 하라
- 후위형(postfix)
  - b++; 우선 원래의 값으로 연산을 한 후 b의 값을 증가시켜라

## 증가 연산자

---

```
#include <stdio.h>
#include <math.h>

void main()
{
 int a=1, b=1;

 printf("a= %d, b= %d\n", ++a, b++);
 printf("a= %d, b= %d\n", a, b);

}
```

---

*Korea University of Technology and Education*

## 관계 연산자, 대입 연산자

---

- <, <= : 작다, 작거나 같다
- >, >= : 크다, 크거나 같다
- ==, != : 같다, 다르다
- a = b : b를 a에 대입
- 혼합 대입 연산자
  - “k=k\*3” 는 “k \*= 3” 와 같다
  - +=, -=, \*=, /=, %=, <<=, >>=, &=, |=

---

*Korea University of Technology and Education*

## 논리 연산자(한 개를 다룸), 비트 연산자 (여러 개를 다룸)

- ! : 논리부정 (negation)
- && : 논리곱 (logical and)
- || : 논리합 (logical or)
- ~ : 비트 부정 (비트논리), 보수 연산
  - $\sim 01011000$   
 $10100111$
  - $\sim a + 1 == -a$  (1의 보수 +1 = 2의 보수 = 음수)
- & : 비트 논리곱 (비트 대 비트 논리곱)
  - $n = n \& 0xfffb$ ; (이진수 1111 1111 1111 1011), 세번째 비트 0
- | : 비트 논리합 (비트 대 비트 논리합)
- ^ : 비트 배타적 논리합 (비트 대 비트 배타적 논리합)
- << : 비트 좌 쉬프트
- >> : 비트 우 쉬프트

*Korea University of Technology and Education*

## 논리 연산자, 비트 연산자

- << 와 >> 사용시 비게 되는 우측 또는 좌측 비트는 0으로 채워진다.

```
int a = 0xc5;
```

```
a <<= 2;
```

```
a : 11000101
```

```
-> 00010100
```

*Korea University of Technology and Education*

# 십진수의 이진형태

```
#include<stdio.h>

void BinPatt(int z);

void main(void)
{
 int a,ok ;

 do {
 printf("Enter a Number to be Binary pattern :Wt");
 ok = scanf("%d",&a); //scanf() 함수는 Ctrl-z 입력시 -1 리턴, EOF 값은 -1로 정의되어 있다.
 BinPatt(a);

 } while(ok != EOF);
}

void BinPatt(int z)
{
 int i;
 for(i = 16; i >=1; i--)
 printf("%d", (z >> i-1) & 0x0001);
 printf("WaWnWn");
}

/*void BinPatt(int z)
{
 int i;
 for(i = 0; i <= 15; i++)
 printf("%d", (z & (0x8000 >> i)) >> 15-i);
 printf("WnWn");
}*/
```

Korea University of Technology and Education

# 조건 연산자

- 변수 = 조건식1 ? 식2: 식3;
  - 조건식이 참이면, 식2의 값을 변수에
  - 조건식이 거짓이면, 식3의 값을 변수에
- $C = (x > 0) ? x : -x;$ 
  - 조건 연산자를 이용해 절대값을 구하는 방법
  - If  $(x > 0)$   $c = x;$   
else  $c = -x$

```
#include<stdio.h>

void main()
{
 int a, abs;
 scanf("%d", &a);
 printf("%dWn", abs = (a > 0)? a : -a);
}
```

Korea University of Technology and Education

## 캐스트 연산자, sizeof

- 강제적으로 형을 변환시켜주는 연산자
  - (float) i; //i를 원래 int로 선언하였어도 이 부분에선 실수형으로 쓰인다.
- 1년은 365.2422일 -> 날, 시간, 분, 초로 나타내어 보자
- sizeof 연산자: 어떤 대상물에 할당된 메모리 크기(byte)를 알려준다
  - sizeof 수식, sizeof (형) -> sizeof(a), sizeof(int)

## 연산자의 종류와 우선순위

| 우선순위 | 연산자의 종류  | 연산자                              |
|------|----------|----------------------------------|
| 1    |          | () [] ->(구조체 멤버연산자)              |
| 2    | 단항 연산자   | *(포인터) &(주소)                     |
| 3    | 산술 연산자   | * / %                            |
| 4    |          | + -                              |
| 5    | 쉬프트 연산자  | << >>                            |
| 6    | 관계 연산자   | < <= > >=                        |
| 7    |          | == !=                            |
| 8    | 비트 연산자   | &                                |
| 9    |          | ^                                |
| 10   |          | !                                |
| 11   | 논리 연산자   | &&                               |
| 12   |          |                                  |
| 13   | 조건 연산자   | ? :                              |
| 14   | 혼합대입 연산자 | = += -= *= /=<br>%=  = ^= &= >>= |
| 15   | 컴마 연산자   | ,                                |

# 분기 제어

- 프로그램의 실행순서를 의도적으로 바꾸어야 할 경우

| 분기문                       | 형식                                                                                                | 내용                                                                                                                                                                                                                                          |
|---------------------------|---------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| if<br>(else if)<br>else 문 | if (조건1) 문장1;<br>else if (조건2) 문장 2;<br>else 문장3                                                  | <pre> graph TD     C1{조건1} -- Y --&gt; M1[문장1]     C1 -- N --&gt; C2{조건2}     C2 -- Y --&gt; M2[문장2]     C2 -- N --&gt; M3[문장3]         </pre>                                                                                              |
| Switch<br>-case문          | switch(S)<br>{case S1:<br>문장 1;<br>break;<br>case S2:<br>문장 2;<br>break;<br>default<br>문장 0;<br>} | <pre> graph TD     S{S} -- S1 --&gt; M1[문장1]     S -- S2 --&gt; M2[문장2]     S -- Sn --&gt; Mn[문장n]     S -- default --&gt; M0[문장0]     M1 --&gt; J(( ))     M2 --&gt; J     Mn --&gt; J     M0 --&gt; J     J --&gt; Exit[ ]         </pre> |

Korea University of Technology and Education

## goto label;

- Break 문이 하나의 루프를 빠져나가는데 비해 이를 이용해 한꺼번에 빠져나갈 수 있다.

Korea University of Technology and Education

# 루프제어

| 루프문      | 형식                                                                                               | 내용 |
|----------|--------------------------------------------------------------------------------------------------|----|
| while    | While (P)<br>{<br>문장;<br>}                                                                       |    |
| do-while | Do {<br>문장;<br>} while()                                                                         |    |
| for      | 초기식 조건식 증감식<br>For(P1; P2; P3) {<br>문장;<br>}<br>Ex) for(i=0;i<10;i++) {<br>문장;}<br>-> 문장을 10번 수행 |    |

Korea University of Technology and Education

# 루프제어

- kbhit() 함수는 키보드가 눌러짐을 검출, conio.h에 선언되어 있음, getch()로 대체하고 결과 비교. "asthree.c"
- 무한루프 방법
  - for( ; ; ) {  
    if(조건) break;  
}
  - while(1) {  
    if (조건) break;  
}
- for문 안에는 함수 등도 들어갈 수 있다. "asc.c"

Korea University of Technology and Education