

1장. C의 개요

1.1 C언어 개요

■ C-Language란?

- ▶ 원하는 결과를 얻어내기 위한 Program 작성시 필요한 일종의 언어
- ▶ Unix 운영 체제하에서 시스템 프로그래밍을 하기위해 개발된 언어
- ▶ 구조적인 언어, 강력한 기능, 빠른 속도

■ C 언어의 역사

- ▶ ALGOL60 (ALGOritmic Language): 1960년 국제 위원회에서 발표
- ▶ CPL (Combined Programing Language) :1963년 영국 캠브리지 런던대학
- ▶ BCLP (Basic CLP) : 영국 캠브리지 런던대학
- ▶ B : 미국 AT&T Bell 연구소의 Ken Thompson, 1970년 발표
- ▶ C : 미국 AT&T Bell 연구소의 Dennis Ritchie, 1972
- ▶ C++ : 표준 C에서 OOP(Object Oriented Programing, 객체지향 프로그램)

의 개념 추가

■ C 언어의 종류

- ▶ ANSI C (American National Standard Institute)
: 표준화 작업, "Standard C"
- ▶ Visual C : Microsoft사에서 개발
- ▶ Borland C : Microsoft사에서 개발

■ 프로그램이 만들어지는 절차

Source Code --> Sorce File --> 선행처리 지시어가 번역 Sorce File
(*.c)

--> Object File --> Executable File
(*.obj) (*.exe)

1.2 프로그래밍 준비 및 실습용 도구

- ▶ DOS or UNIX 편집기, Wordprocessor
- ▶ MS-DOS 호환체제가 설치된 XT급 이상의 PC
C-compiler가 설치된 Unix System
- ▶ Tool : Turbo-C 2.0 이상, Borland C, Visual C++

1.3 C-Program의 기본적인 구조

■ 예제1) source1-1.c (화면출력)

```
#include <stdio.h>
int main(void)
{
    printf("from sea to shining C \n ");
    return 0;
}
```

==> from sea to shining C

▶ #include <stdio.h>

컴파일 전 전처리기 수행, printf 함수 사용 위해

<stdio.h> 헤더 파일 복사

Ⓡ 필요하다면 헤더파일을 지정한다

▶ int main(void)

main() 함수 정의, int, void -> key word,

int형 return 값을 갖는 함수

Ⓡ 프로그램은 함수로 구성된다.

Ⓡ 프로그램에는 하나의 main() 함수가 필요하다.

▶ { }

함수정의 본체(body)

Ⓡ 함수는 { 로 시작하고 }로 끝난다.

▶ printf()

화면출력기능의 Library 함수, stdio.h

ⓐ printf()는 함수이다.

▶ “from sea to shining C \n ”

“ ... 출력내용 ...”

Ⓜ 개행(new line)은 프로그래머가 지시한다.

▶ printf("from sea to shining C \n ");

Ⓜ 문장 끝에는 ;(세미콜론)이 필요하다

▶ return 0;

0값을 운영체제에 return

▶ source1-1.c과 결과가 같은 program

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

```
int main(void)
```

```
{
```

```
    printf("from sea to");
```

```
    printf("shining C ");
```

```
    printf(" \n ");
```

```
    return 0;
```

```
}
```

==> from sea to shining C

▶ source1-1.c의 결과를 3줄로 출력하는 program

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

```
int main(void)
```

```
{
```

```
    printf("from sea ");
```

```
    printf("to shining \n C \n ");
```

```
    return 0;
```

```
}
```

==> from sea
to shining
C

- ▶ source1-1.c의 결과를 *로 만든 사각형 안에 문구가 출력되도록하는 program

```
#include <stdio.h>
int main(void)
{
    printf("*****\n");
    printf("*      from sea      *\n");
    printf("*      to shining C    *\n");
    printf("*****\n");
    return 0;
}
```

```
==> *****
      *      from sea      *
      *      to shining C  *
      *****
```

- ▶ source1-1.c과 결과가 같은 program

```
#include <stdio.h>
int main(void)
{printf("from sea to shining C \n "); return 0;}
```

==> from sea to shining C

☐ C에는 행의 개념이 없다.

1.4 변수, 수식, 배정

```
/* The distance of a marathon in kilometers. */
#include <stdio.h>
int main(void)
{
    int      miles, yards;
    float    kilometers;

    miles=26;
```

```

yards=385;
kilometers=1.609*(miles + yards / 1760.0 );
printf("\n A marathon is %f kilometers. \n\n", kilometers);
return 0;
}

```

==> A marathon is 42.185970 kilometers.

- ▶ `/* The distance of a marathon in kilometers. */`
☐ 주석문은 `/*` 와 `*/` 로 둘러싼다
- ▶ `int miles, yards;`
선언문, 변수 `miles`, `yards`가 정수값을 가지는 변수
- ▶ `float kilometers;`
선언문, 변수 `kilometers`가 실수값을 가지는 변수
☐ 변수는 선언하고 나서 사용한다
- ▶ `miles=26;`
`yards=385;`
배정문, 정수형 상수 26과 385가 변수 `miles`과 `yards`에 각각 배정된다.
- ▶ `kilometers=1.609*(miles + yards / 1760.0);`
배정문,
`/`, `*`, `+` : 연산자 (`-`, `%`, ...)
- ▶ `printf("\n A marathon is %f kilometers. \n\n", kilometers);`
변환형식 `%f` 과 인자 `kilometers`는 짝을 이루며, `kilometers`의 값이 부동 소수점(float) 형식 `%f`의 위치에 출력됨.
☐ 변수의 값을 출력하려면 서식지정이 필요하다.
- ▶ 수식의 변환 규칙 (Conversion rule)
`7/2` --> 3, `7.0/2` --> 3.5

1.5 #define 과 #include

```

#define LIMIT 100
#define PI 3.14159
#define C 299792.458 /* speed of light in km/sec */
▶ # : 전처리 지시자 (Preprocessing Directive)

```

```
#include "my_file.h"
```

▶ file에 my_file.h 파일의 사본 포함

▶ C에서 제공하는 표준 헤더파일 : stdio.h, string.h, math.h, <xxx.h>

In file pacific_sea.h;

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

```
#define AREA 2337
```

```
#define SQ_MILE_PER_SQ_KILOMETER 0.386102
```

(이하 TEXT P19)

▶ #define SQ_FEET_PER_SQ_MILE (5280*5280)

▶ const int pacific_sea = AREA; /* in sq km */

const : ANSI C 에 소개된 한정자

▶ float : 유효숫자 6자리, double :16자리

▶ printf("%22.7e acres\n", acrs);

5.7748528e+05 acres

%m.ne : m개 자릿 내에 소숫점의 오른쪽으로 n개의
숫자를 가진 지수형태

1.6 printf() 와 scanf()의 사용

▶ printf() : 화면 출력

printf(“서식지정문자열“, ”변수“);

▶ 서식지정 문자열

일반문자열, 변환문자열(%), 확장문자열(\)

▶ printf()의 변환문자열

printf(“%변환문자“, ”변수“);”변수“를 변환형식에 맞추어 화면 출력

▶ 변환문자 (Conversion Character) - printf()

| | |
|----|---|
| c | as a character (문자) |
| d | as a decimal integer (10진 정수) |
| ld | as a long type decimal integer (long형 10진 정수) |
| e | as a floating point number in scientific notation (지수형..) |
| f | as a floating point number (float, double) |
| g | in the e-format or f-format, whichever is shorter |
| s | as a string (문자열) |

▶ 화면에 abc 출력

printf("abc");

printf("%s", "abc");

printf("%c%c%c", 'a','b','c');

▶ printf()의 옵션 지정

| % | 필드 폭 | 자릿수 | 변환문자 |
|-------|------|----------------|--------------------------------|
| %d | -> | 123 | |
| %5d | -> | _ _123 | |
| %10d | -> | _ _ _ _ _ _123 | |
| %2d | -> | 123 | (지정폭이 작아도 필요한 폭은 확보) |
| %f | -> | 654.321000 | (표준폭으로 출력) |
| %12f | -> | _ _ 654.321000 | (소수점 넣어 12자리로출력, 이하는 표준폭으로 출력) |
| %9.2f | -> | _ _ _ 654.32 | (소수점 넣어 9자리로출력, 이하는 2자리로 출력) |

▶ scanf() : 키보드 입력

▶ scanf()의 변환문자열

printf("%변환문자", &변수); 변환문자 형식으로 입력 받아들임.

▶ 변환문자 (Conversion Character) - scanf()

| | |
|----|---|
| c | to a character (문자) |
| d | to a decimal integer (10진 정수) |
| ld | to a long type decimal integer (long형 10진 정수) |
| f | to a floating point number (float) |
| lf | to a floating point number (double) |
| LF | to a floating point number (long double) |
| s | to a string (문자열) |

▶ 예제 프로그램

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

```
int main(void)
```

```
{
```

```
    char        c1,c2,c3;
```

```
    int          i;
```

```
    float        x;
```

```
    double       y;
```

```
    printf("\n%s\n%s", "Input three character,"
```

```
           "an int, a float, and a double: ");
```

```
    scanf("%c%c%c%d%f%lf", &c1, &c2, &c3, &i, &x, &y);
```

```
    printf("\nHere is the data that you typed in:\n");
```

```
    printf("%3c%3c%3c%5d%17e%17e\n\n", c1,c2,c3,i,x,y);
```

```
}
```

```
입력 ==> ABC 3 55 77.7
```

```
==>Input three character,
```

```
an int, a float, and a double: ABC 3 55 77.7
```

```
Here is the data that you typed in:
```

```
_ _ A_ _ B_ _ C_ _ _ _3_ _ _ _ _ 5.500000e+01_ _ _ _ _  
7.770000e+01
```


1.7 제어의 흐름

- ▶ 선택 : if, if-else
- ▶ 반복 : for, while, do-while

■ if 문

- ▶ 일반적인 형태 : if (조건식) { 문장1 }
-> 조건식이 참(true)이면 (0이 아니면) 문장1 실행, 단문이면 {} 생략
- ▶ a=1
if (b==3) a=5; /* == : '--와 같다' 연산자 */
printf("%d", a);
-> b가 3이면 a=5, (b==3) ->1 (참)
-> b가 3이 아니면 문장(a=5) 실행 안함, printf() 문 실행 1 출력
(b==3) ->0 (거짓)

■ if-else 문

- ▶ 일반적인 형태 : if (조건식) { 문장1 }
else { 문장2 }
-> 조건식이 참(true)이면 (0이 아니면) 문장1 실행
그렇지 않으면 문장2 실행

▶ 예문

```
if(cnt==0){  
    a=2;  
    b=3;  
    c=5;  
}  
else {  
    a=-1;  
    b=-2;  
    c=-3;  
}  
printf("%d", a+b+c);
```

==> cnt 가 0값을 가지면 10 출력, 그렇지 않으면 -6 출력

■ while loop

```
#include <stdio.h>

int main(void)
{
    int i=1, sum=0;

    while (i<=5) {
        sum+=i;
        ++i;
    }
    printf("sum= %d \n", sum);
    return 0;
}
```

▶ while 문(Loop)

```
while (i<=5) {
    sum+=i;
    ++i;
}
```

+= : 작거나 같음.

▶ sum+=i;

sum=sum+i;

▶ ++i; 증가

i=i+1;

--i; 감소

i=i-1;

■ for loop

```
#include <stdio.h>

int main(void)
{
    int sum=0;

    for (i=1; i<=5; ++i ) {
        sum+=i;
    }
    printf("sum= %d \n", sum);
    return 0;
}
```