15. C언어의 핵심 - 함수

15-1. 함수

C언어는 함수로 시작해서 함수로 끝이 난다.

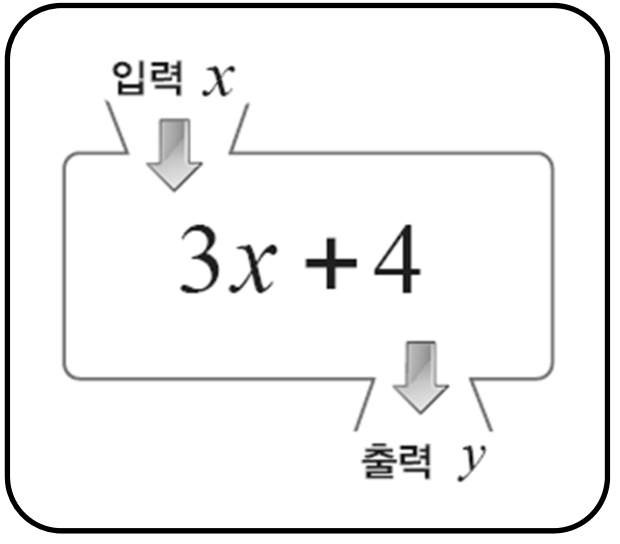
C 프로그래밍의 실력은 함수를 어떻게 만드느냐에 따라 좌우된다.

15-2. 함수의 정의와 선언

- **수학적 개념의 함수와 C 언어의 함수**

[수학적 개념]

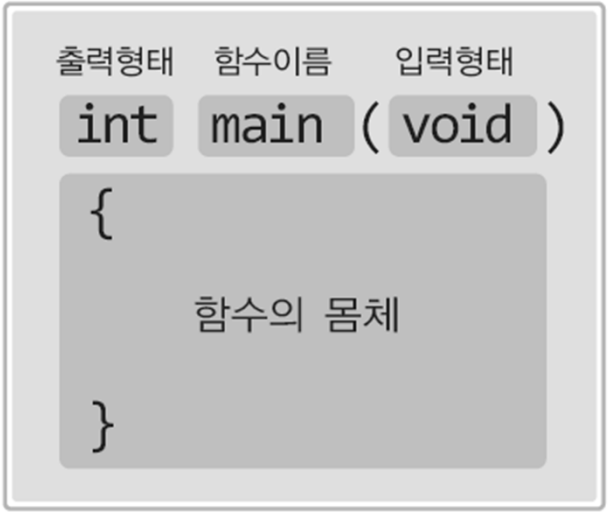
y = 3x + 4



[프로그램 개념]

```
int sum(char x)
{
    y=3x+4;
    return y;
}
```

- **함수의 구조**



· **함수의 이름** : 함수를 호출할 때 사용하는 이름

· **출력형태** : 실행의 결과

· **입력형태** : 함수 호출 시, 전달되는 입력 값.

15-2. 함수의 정의와 선언

함수를 정의 하는 이유

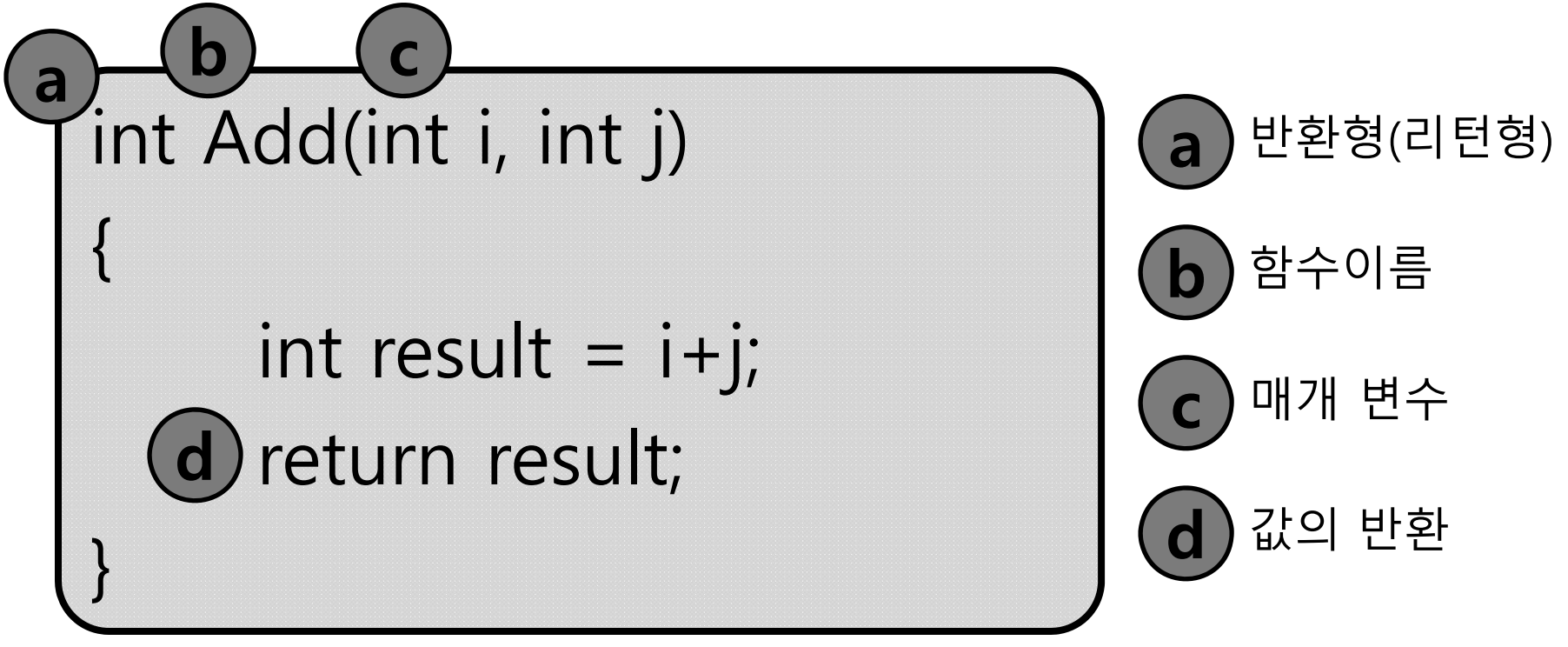
- 소스코드의 중복성을 없애 준다.
- 한번 제작된 함수는 다른 프로그램 제작 때도 사용이 가능하다.
- 복잡한 문제를 단순한 부분으로 분해할 수 있다.

4 가지 형태의 함수 (전달 인자와 반환 값에 의해 결정)

전달인자(입력) O , 반환값(출력) O	int Add(int i, int j)
전달인자(입력) O , 반환값(출력) X	
전달인자(입력) X , 반환값(출력) O	
전달인자(입력) X , 반환값(출력) X	

15-2. 함수의 정의와 선언

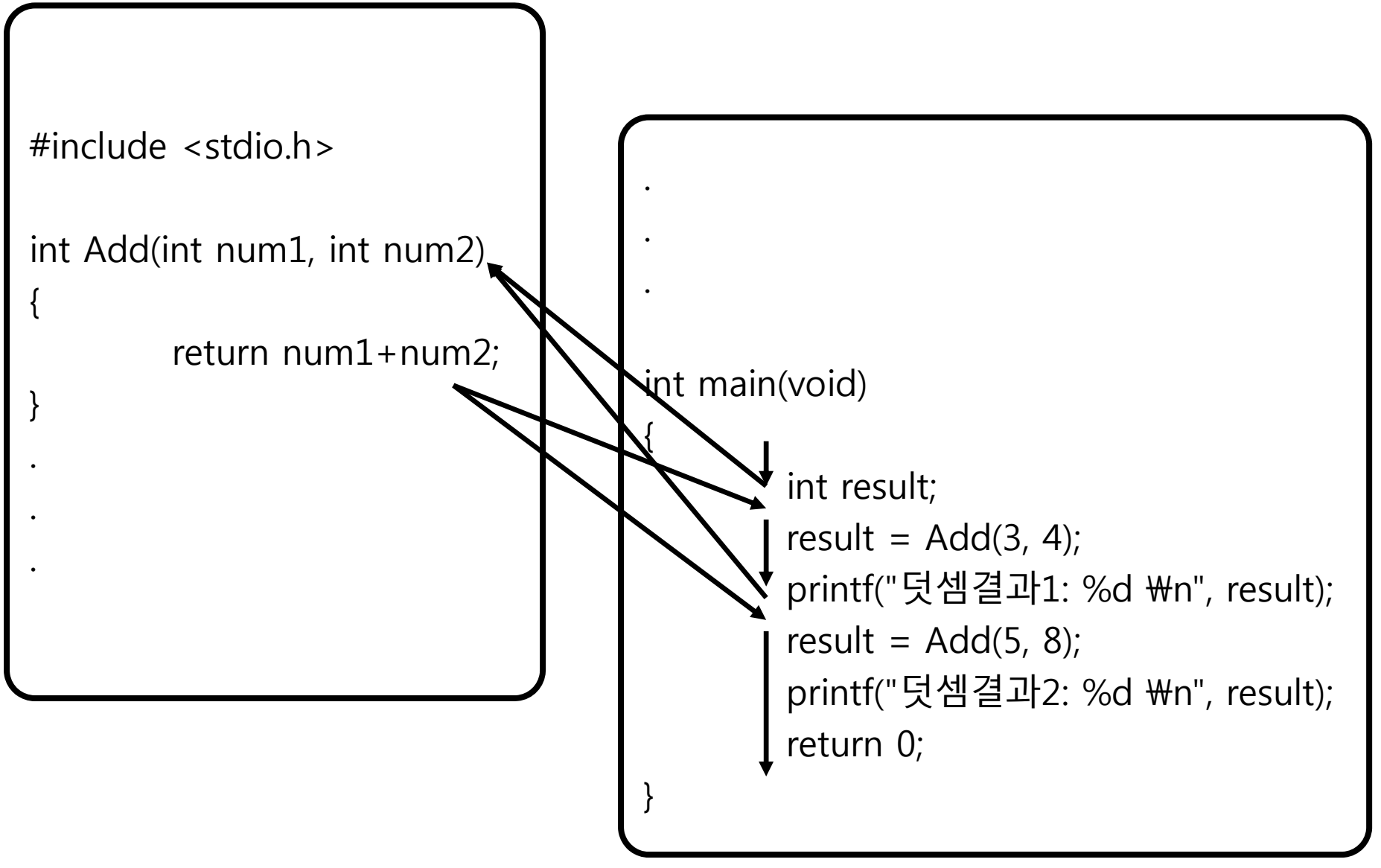
- 전달 인자(입력) O, 반환 값(출력) O



15-2. 함수의 정의와 선언 (예제 15-1)

```
#include <stdio.h>

int Add(int num1, int num2)
{
        return num1+num2;
}
int main(void)
{
        int result;
        result = Add(3, 4);
        printf("덧셈결과1: %d ₩n", result);
        result = Add(5, 8);
        printf("덧셈결과2: %d ₩n", result);
        return 0;
}
```

15-2. 함수의 정의와 선언 (예제 15-1)



15-2. 함수의 정의와 선언

- **전달 인자(입력) O, 반환 값(출력) X**

```
void ShowAddResult(int num)
{
        printf("덧셈결과 출력:%d₩n", num);
}
```

- **전달 인자(입력) X, 반환 값(출력) O**

```
int ReadNum(void)
{
        int num;
        scanf("%d", &num);
        return num;
}
```

15-2. 함수의 정의와 선언

- **전달 인자(입력) X, 반환 값(출력) X**

```
void HowToUseThisProg(void)
{
	printf("두 개의 정수를 입력하시면 덧셈결과가 출력됩니다.₩n");
	printf("자! 그럼 두 개의 정수를 입력하세요.₩n");
}
```

15-2. 함수의 정의와 선언(예제 15-2)

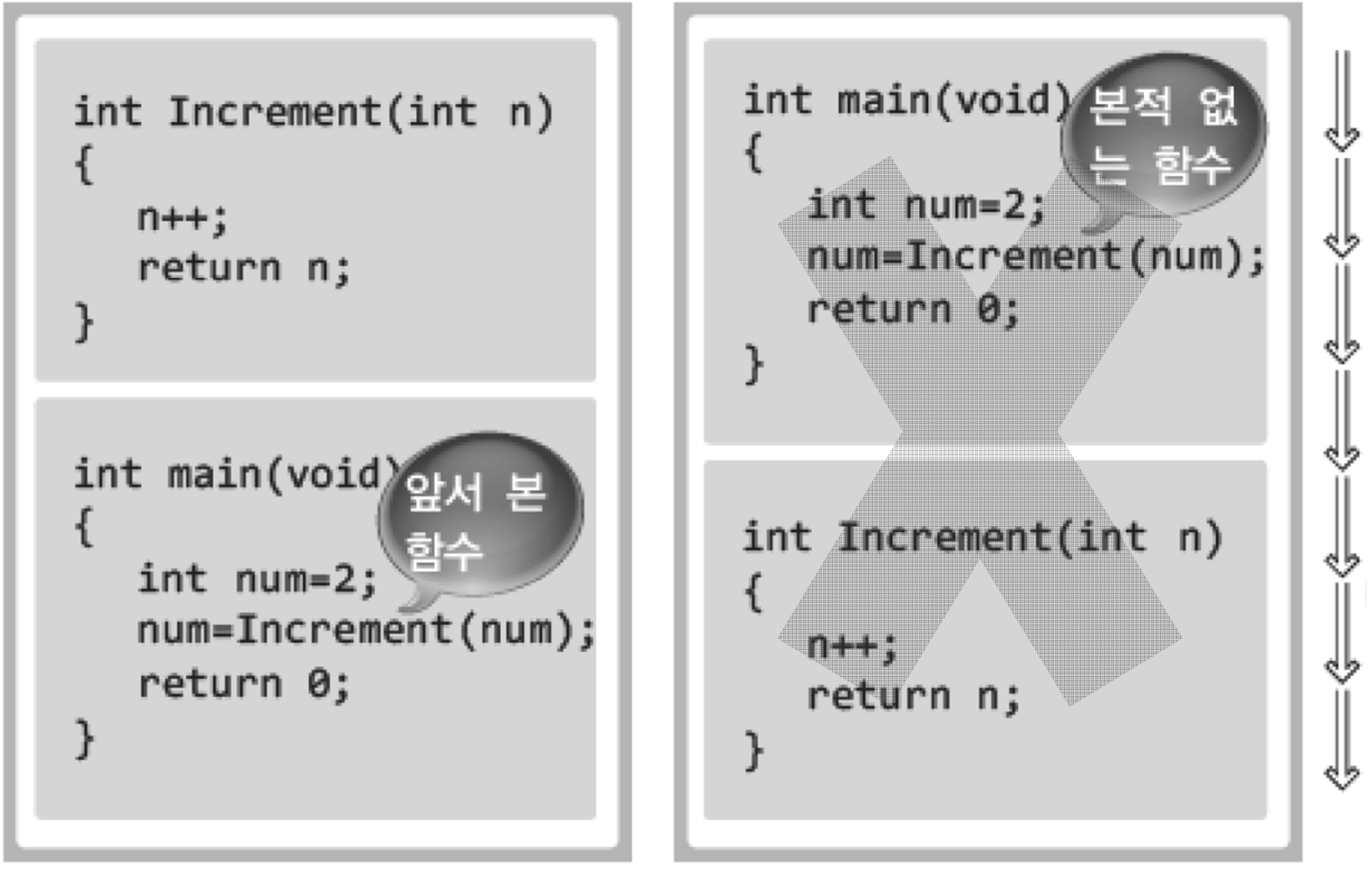
```
#include <stdio.h>

int Add(int num1, int num2)      // 인자전달 (O), 반환 값 (O)
{
        return num1+num2;
}
void ShowAddResult(int num)      // 인자전달 (O), 반환 값 (X)
{
        printf("덧셈결과 출력: %d ₩n", num);
}
int ReadNum(void)      // 인자전달 (X), 반환 값 (O)
{
        int num;
        scanf("%d", &num);
        return num;
}
```

15-2. 함수의 정의와 선언(예제 15-2)

```
void HowToUseThisProg(void)      // 인자전달 (X), 반환 값 (X)
{
        printf("두 개의 정수를 입력하시면 덧셈결과가 출력됩니다. ₩n");
        printf("자! 그럼 두 개의 정수를 입력하세요. ₩n");
}
int main(void)
{
        int result, num1, num2;
        HowToUseThisProg();
        num1=ReadNum();
        num2=ReadNum();
        result = Add(num1, num2);
        ShowAddResult(result);
        return 0;
}
```

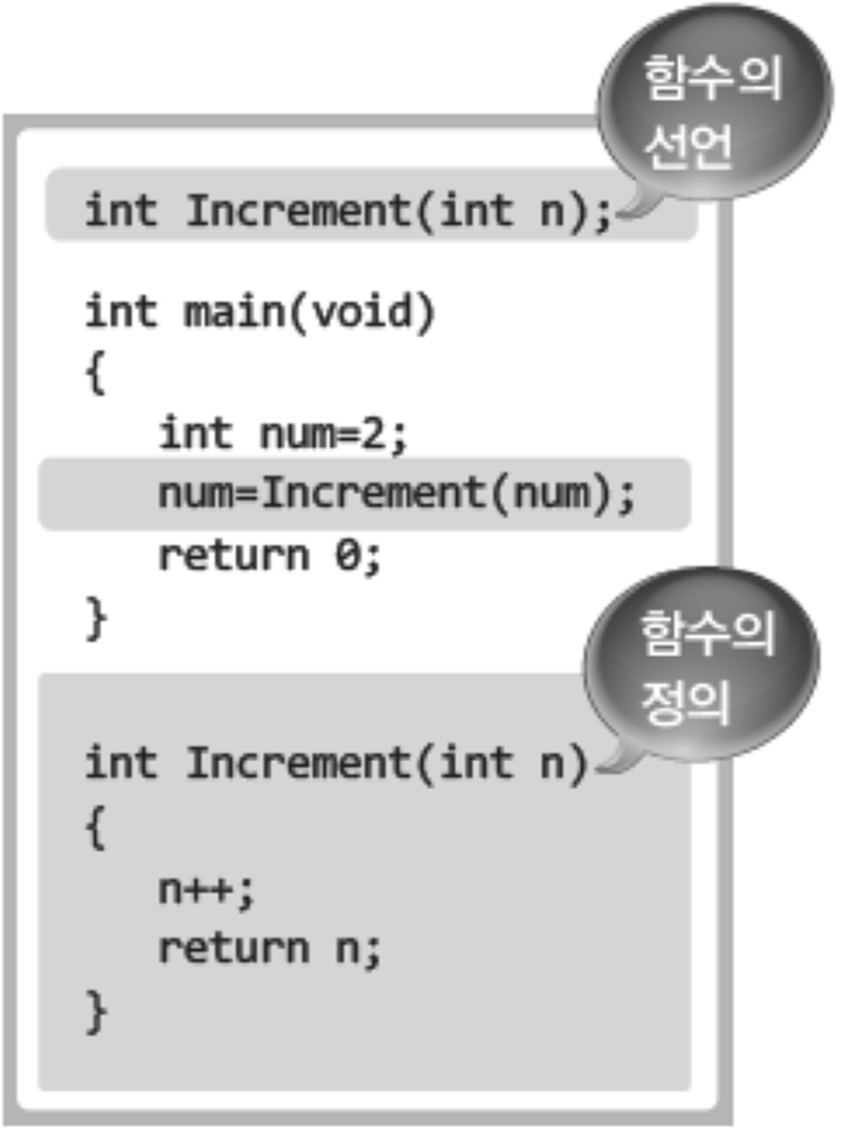
15-2. 함수의 정의와 선언

- 함수의 선언



15-2. 함수의 정의와 선언

- 함수의 선언



15-2. 함수의 정의와 선언(예제 15-3)

```
#include <stdio.h>
int NumberCompare(int , int );

int main(void)
{
        printf("3과 4중에서 큰 수는 %d 이다. ₩n", NumberCompare(3, 4));
        printf("7과 2중에서 큰 수는 %d 이다. ₩n", NumberCompare(7, 2));
         return 0;
}
int NumberCompare(int num1, int num2)
{
        if(num1>num2)
                return num1;
        else
                return num2;
}
```

15-2. 함수의 정의와 선언(예제 15-4)

```
#include <stdio.h>
int AbsoCompare(int , int );  // 절대값이 큰 정수 반환
int GetAbsoValue(int );   // 전달인자의 절대값을 반환

int main(void)
{
        int num1, num2;
        printf("두 개의 정수 입력: ");
        scanf("%d %d", &num1, &num2);
        printf("%d와 %d중 절대값이 큰 정수: %d ₩n",
                num1, num2, AbsoCompare(num1, num2));
        return 0;
}
```

15-2. 함수의 정의와 선언(예제 15-4)

```
int AbsoCompare(int num1, int num2)
{
	if(GetAbsoValue(num1) > GetAbsoValue(num2))
		return num1;
	else
		return num2;
}

int GetAbsoValue(int num)
{
	if(num<0)
		return num * (-1);
	else
		return num;
}
```

15-3. 라이브러리 함수

- **함수는 사용자 정의 함수와 라이브러리 함수로 나뉜다.**

  표준라이브러리 함수 : 컴파일러에서 제공하는 함수

  - **표준 입출력** : scanf( ), printf( ) 등등
  - **수학연산**
  - **문자열 처리**
  - **시간 처리**
  - **오류 처리**
  - **데이터 검색과 정렬**

15-3. 라이브러리 함수(예제 15-5)

```
#include <stdio.h>

int main(void)
{
        int   i;

        for(i = 0 ; i < 6 ; i ++ )
            printf("%d ", rand());
        return 0;
}
```

15-3. 라이브러리 함수(예제 15-6)

```
#include <stdio.h>

int main(void)
{
        int   i;

        for(i = 0 ; i < 6 ; i ++ )
            printf("%d ", 1+(rand() % 45));  // 1부터 45까지의 난수
        return 0;
}
```

15-3. 라이브러리 함수(예제 15-7)

```
#include <stdio.h>

#define MAX 45

int main(void)
{
  int i;
  srand((unsigned) time(NULL) );

 // 로또 번호 발생기
  for( i = 0 ; i < 6; i ++)
    printf("%d ", 1+rand()%MAX );

   return 0 ;
}
```

15-3. 라이브러리 함수(예제 15-8)

```
#include <stdio.h>

int coin_toss(void);
int main(void)
{
        int toss;
        int heads = 0;
        int tails = 0;
        srand((unsigned) time(NULL));

        for(toss = 0; toss <100 ;toss++){
                if(coin_toss()==1) heads++;
                else tails++;
        }
        printf("동전의 앞면: %d ₩n",heads);
        printf("동전의 뒷면: %d ₩n",tails);

        return 0;
}
```

15-3. 라이브러리 함수(예제 15-8)

```
int coin_toss(void)
{
        int i ;

        i = rand() %2 ;
        if(i==0)
            return 0;
        else
            return 1;
}
```

15-3. 라이브러리 함수

수학 함수

분류	함수	설명
삼각함수	double sin(double x)	사인 값 계산
	double cos(double x)	코사인 값 계산
	double tan(double x)	탄젠트 값 계산
기타함수	double ceil(double x)	x보다 작지 않은 가장 작은 정수
	double floor (double x)	x보다 크지 않은 가장 큰 정수
	int abs(int x)	정수 x의 절대값
	double fabs(double x)	실수 x의 절대값
	double pow(double x, double y)	x^y
	double sqrt(double x)	$\sqrt{x}$

15-3. 라이브러리 함수 radian 값 =

- sin, cos, tan 함수는 radian을 사용하여 계산한다.

$$\text{radian 값} = \frac{\Pi}{180} \times x^{o}$$

- ceil(double x)

ceil(1.8) ; // 2.0을 반환

ceil(-1.8) ; // -1.0을 반환

- floor(double x)

floor(1.8) ; // 1.0을 반환

floor(-1.8) ; // -2.0을 반환

15-3. 라이브러리 함수 radian 값 =

- **abs(int x), fabs(double x)**

  abs(-7) ; // 7 을 반환

  fabs(-2.87) ; // 2.87을 반환

- **pow(double x, double y)**

  pow(2.0, 3.0) ; // 8.0을 반환

- **sqrt(double x)**

  sqrt(16.0) ; // 4.0을 반환

15-3. 라이브러리 함수(예제 15-9)

```
#include <stdio.h>
#include <math.h>
int main(void)
{
        double pi = 3.141592;
        double radian, x;

        printf(" sin, cos, tan 각도를 입력하시오:");
        scanf("%lf",&x);

        radian =  (pi/ 180) * x ;

        printf("sin(x)= %f  cos(x)= %f tan(x) = %f₩n",
                sin(radian),cos(radian),tan(radian));
        return 0;
}
```

15-3. 라이브러리 함수(예제 15-10)

```
#include <stdio.h>
#include <math.h>
int main(void)
{
        double c1, c2, f1, f2 ;

        c1 = ceil(2.5);
        c2 = ceil(-2.5);

        f1 = floor(2.5);
        f2 =floor(-2.5);

        printf("c1 = %f c2 = %f f1 = %f f2 = %f ",
                c1, c2, f1, f2);
        return 0;
}
```

15-3. 라이브러리 함수(예제 15-11)

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

int main(void)
{
        int a=0, b=0 ;
        double da=0.0, db=0.0 ;

        a = -9 ;
        b= abs(a);

        da = -19.87 ;
        db = fabs(da);

        printf("b = %d  db = %f ₩n",b,db);
        return 0;
}
```

15-3. 라이브러리 함수(예제 15-12)

```
#include <stdio.h>
#include <math.h>
int main(void)
{
        double x, y, z;

        x = 2.0;
        y = 3.0;

        z= pow(x,y);
        printf("z = %f₩n",z);

        x = 9.0;
        y=sqrt(x);

        printf("y = %f₩n",y);
        return 0;
}
```

15. 연습문제

1. 예제 15-10에서 배웠던 ceil(double x) 함수와 같은 기능의 함수를 만들어 보시오.

(함수 이름은 make_ceil(double x)로 하시오.)

실행 예) make_ceil(2.5) // 3.0을 반환
make_ceil(-2.5) // -2.0을 반환

2. 예제 15-10에서 배웠던 floor(double x) 함수와 같은 기능의 함수를 만들어 보시오.

(함수 이름은 make_floor(double x)로 하시오.)

실행 예) make_floor(2.5) // 2.0을 반환
make_floor(-2.5) // -3.0을 반환

15. 연습문제

3. 예제 15-11에서 배웠던 abs(int x) 함수와 같은 기능의 함수를 만들어 보시오.
(함수 이름은 make_abs(int x)로 하시오.)

실행 예) make_abs (-7) // 7을 반환
make_abs(7) // 7을 반환

4. 예제 15-11에서 배웠던 fabs(double x) 함수와 같은 기능의 함수를 만들어 보시오.
(함수 이름은 make_fabs(double x)로 하시오.)

실행 예) make_fabs(-3.14) // 3.14를 반환
make_fabs(3.14) // 3.14를 반환

15. 연습문제

5. 예제 15-12에서 배웠던 pow(int x, int y) 함수와 같은 기능의 함수를 만들어 보시오.
(함수 이름은 make_pow(int x, int y)로 하시오.)

실행 예제) make_pow(2,10) // 1024.00을 반환
make_pow(2, -1) // 0.5 를 반환

6. 다음과 같은 간단한 기능을 하는 함수들을 작성하고, 사용자로부터 임의의 값을 입력 받은 후에 작성한 함수들을 테스트하여 보자.

6-1. 주어진 정수가 짝수이면 1을 반환하고 홀수 이면 0을 반환하는 함수
int even(int n)
6-2. 주어진 정수가 음수이면 -1을, 양수이면 1을 0이면 0을 반환하는 함수
int sign(int n)

15. 연습문제

7. 화씨 온도(F)를 섭씨 온도(C)로 변환하는 함수 f_to_c(double f)를 작성한다. 다음과 같은 식을 이용하라. 사용자로 부터 화씨 온도를 입력 받아서 함수를 호출하여 섭씨 온도로 변환한 후에 화면에 출력하도록 하자.

   C = 5.0 /9.0 * (F -32.0)

8. 앞장에서 공부한 윤년을 구하는 알고리즘을 이용하여 함수 is_leap(int year) 를 작성하고 이 함수를 사용하여서 1년이 몇 일인지를 출력하는 프로그램을 작성하여 보자

   실행 예) 연도를 입력하시오 : 2012
   2012년은 366일 입니다
   (윤년은 4의 배수이지만 100의 배수는 제외하고, 400의 배수는 추가한다. 윤년은 366일이고, 평년은 365일이 된다)

15. 연습문제

9. 두 점사이의 거리를 계산하는 함수를 작성하여 보자. 2차원 공간에서 두 점 $(x_1,\ y_1)$와 $(x_2,\ y_2)$ 사이의 거리를 계산하는

get_distance(double x1, double y1, double x2, double y2)를 작성하라.
다음과 같은 두 점 사이의 거리를 계산하는 공식을 사용하라.
제곱근은 sqrt( ) 라이브러리 함수를 사용하라.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

실행 예)

첫 번째 점의 좌표를 입력하시오: (x, y) 1 1
두 번째 점의 좌표를 입력하시오: (x, y) 10 10
두 점사이의 거리는 12.727922 입니다.

15. 연습문제

10. 입력되는 두 정수 사이의 정수의 합을 구하는 함수 is_sum(int x1, int x2) 을 작성하시오.

실행 예)
두 정수를 입력하시오 : 1 10
두 정수 사이의 정수의 합은 55 입니다.

11. 입력되는 두 수사이의 정수들 가운데 소수들을 구하는 함수 is_prime(int x1, int x2) 을 작성하시오.

실행 예)
두 정수를 입력하시오 : 1 10
2 3 5 7

15. 연습문제

12. 세 개의 매개변수 인자를 받는 함수를 만드시오. 세 개의 변수 중 첫 번째 매개변수는 '+', '-', '*', '/' 연산자를 입력할 수 있도록 하고, 나머지 두 개의 매개변수는 실수를 받아 첫 번째 매개변수에 따라 연산을 하도록 하는 함수를 작성 하시오.
(함수 이름은 double is_cal(char op, double x, double y) 로 한다)

실행 예)
연산자를 입력하시오 : +
두 개의 실수를 입력하시오 : 57.1 32.3
결과 값 = 89.3

15. 연습문제

13. 세 개의 정수 매개변수 인자를 받아 최대값을 구하여 리턴하는 함수를 int max(int x1, int x2, int x3) 와 최소값을 리턴하는 함수 int min(int x1, int x2, int x3)를 작성하고, 최대값과 최소값을 출력하는 프로그램을 작성하시오

실행 예)

세 개의 정수를 입력하시오 : -10 115 33
최대 값 = 115
최소 값 = -10

15. 연습문제

14. 하나의 영문자(alphabet)를 입력 받아 대문자는 소문자로, 소문자는 대문자로 변환하는 프로그램을 작성하시오. 대문자, 소문자 외의 문자는 그대로 출력하도록 하시오.
    함수의 이름은 char is_alph(char t) 로 하시오.

실행 예)
영문 대문자, 또는 소문자를 입력하시오 : a
결과 값 : A