WEEK 05

조건과 반복

학습목표

I. 제어문과 조건문 이해 및 구현
II. 다양한 선택 switch 문 이해 및 구현
III. 다양한 반복문 이해 및 구현

학습목차

Ⅰ. 제어문과 조건문

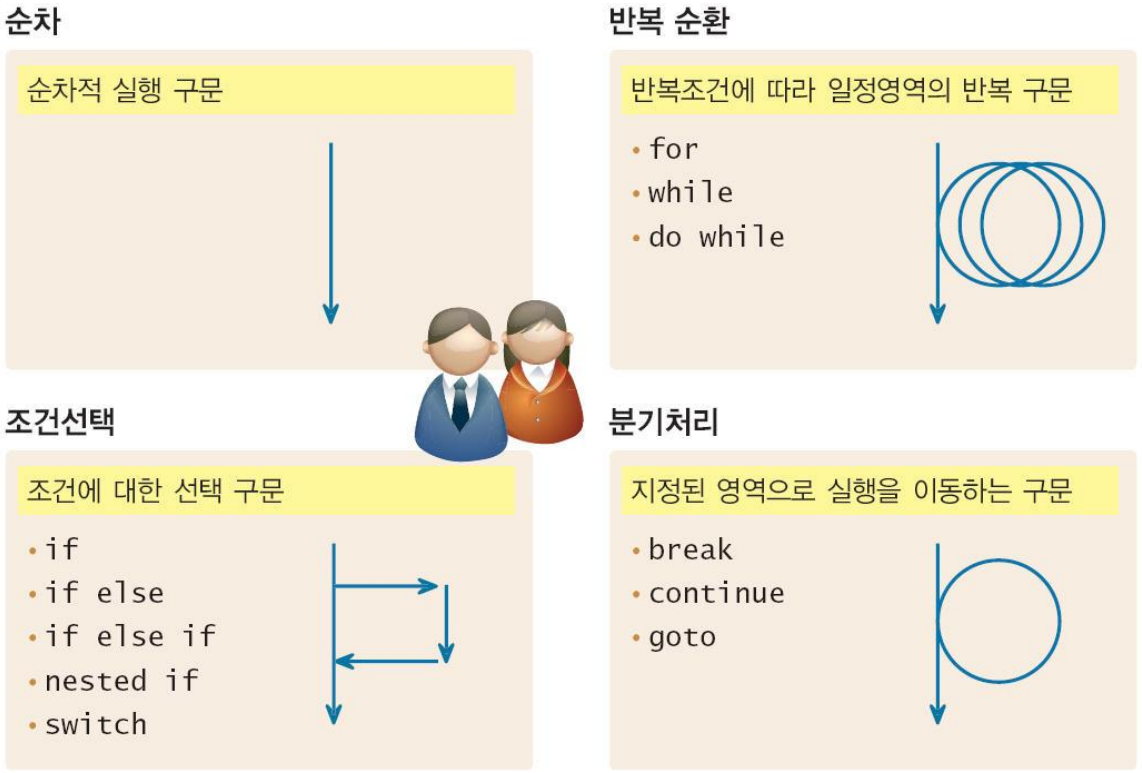
1. 제어문
2. if 문
3. 중첩된 if 문
4. 다양한 if 문의 이용과 조건연산자

1. 제어문

❖ 순차적 실행

❖ 제어문(control statement)

- 조건 선택, 반복(순환), 분기처리



[출처] 강환수 외, Perfect C 3판, 인피니티북스

2. if 문

◆ 조건에 따른 선택의 사례

평균평점 >= 3.5

대학 A는 평균평점이 3.5는 넘어야 장학금을 받을 수 있다.

석차 <= 0.05 * 학생수

대학 B는 학과 석차가 상위 5% 이어야 장학금을 받을 수 있다고 한다.

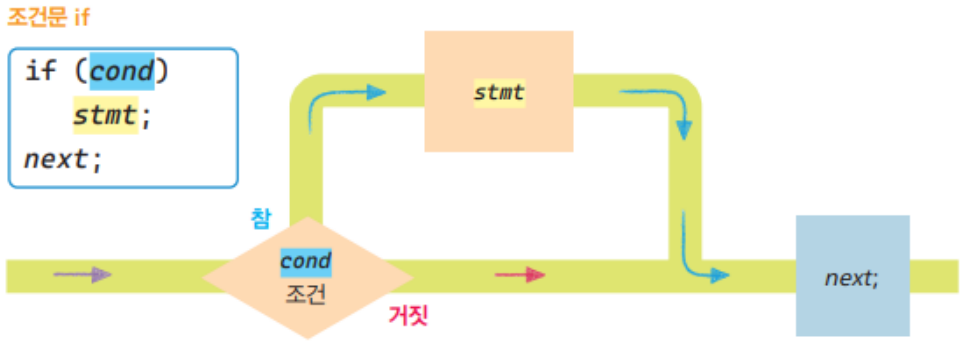
조건 선택의 예	기준 변수	조건 표현의 의사코드
온도가 30도 이상이면 "폭염 주의"를 출력	온도 temperature	만일 (temperature >= 30) printf("폭염 주의");
낮은 혈압이 90이상이면 "고혈압 초기"로 진단	혈압 low_pressure	만일 (low_pressure >= 90) printf("고혈압 초기");
속도가 40km와 60km 사이이면 "적정 속도"라고 출력	속도 speed	만일 (40 <= speed && speed <= 60) printf("적정 속도");

운전면허 필기시험에서 60점 이상이면 "합격", 아니면 "불합격" 출력	시험 성적
	point
		만일 (point >= 60)
		  printf("면허시험 합격");
		아니면
		  printf("면허시험 불합격");

남성일 경우 체력 테스트에서 80점 이상이면 "합격"이고, 아니면 "불합격", 여성이면 70점 이상이면 "합격", 아니면 "불합격"	성별
	type
	
	체력 점수
	point
		만일 남성이면 (type == 1)
		  만일 (point >= 80)
		  printf("남성: 합격");
		  아니면
		  printf("남성: 불합격");
		아니고 만일 여성이면 (type == 2)
		  만일 (point >= 70)
		  printf("여성: 합격");
		  아니면
		  printf("여성: 불합격");

[출처] 강환수 외, Perfect C 3판, 인피니티북스

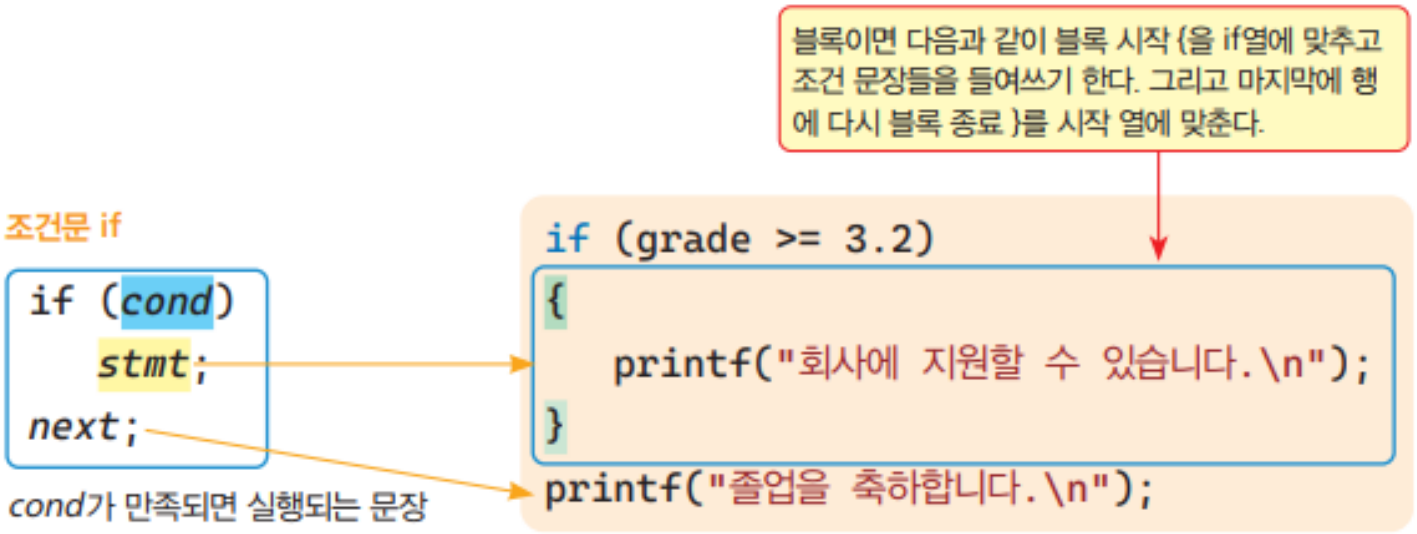
2. if 문



◆ 조건에 따른 선택 if

❖ if

- if 문에서 조건 cond가 0이 아니면(참) stmt를 실행하고,
- 0이면(거짓) stmt를 실행하지 않음



[출처] 강환수 외, Perfect C 3판, 인피니티북스

2. if 문

◆ if문의 주의점

❖ 조건 연산식 이후의 ;

- 조건식 다음에 세미콜론을 바로 써서는 논리 오류
- 즉 if 조건을 만족하지 않아도 stmt;가 항상 실행되는 논리 오류가 발생

왼쪽은 오른쪽과 같은 소스이며 학점이 4.0 이상이라도 실행되는 문장이 하나도 없는 if문이 된다.
그러므로 다음 두 문장은 항상 실행되는 결과를 낳는다.

```
if (grade >= 4.0);
   printf("삼성에 지원할 수 있습니다.\n");

printf("졸업을 축하합니다.\n");
```

==
동일한 소스

```
if (grade >= 4.0);

printf("삼성에 지원할 수 있습니다.\n");
printf("졸업을 축하합니다.\n")
```

[출처] 강환수 외, Perfect C 3판, 인피니티북스

실습예제 6-1 | Prj01 | 01basicif.c | 현재 온도에 따른 폭염 주의 발령 | 난이도: ★

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main(void)
{
   double temperature;

   printf("현재 온도 입력: ");
   scanf("%lf", &temperature);

   if (temperature >= 30.0)
   {
      printf("폭염 주의보를 발령합니다.\n");
      printf("건강에 유의하세요.\n");
   }
   printf("현재 온도는 섭씨 %.2f 입니다.\n", temperature);

   return 0;
}
```

다음과 같이 블록 {의 시작을 조건식 오른쪽에 작성 하기도 함

```c
if (temperature >= 32.0) {
   printf("폭염 주의보를 발령합니다.\n");
   printf("건강에 유의하세요.\n");
}
```

결과

```
현재 온도 입력: 29.3
현재 온도는 섭씨 29.30 입니다.
```

```
현재 온도 입력: 34.678
폭염 주의보를 발령합니다.
건강에 유의하세요.
현재 온도는 섭씨 34.68 입니다.
```

[출처] 강환수 외, Perfect C 3판, 인피니티북스

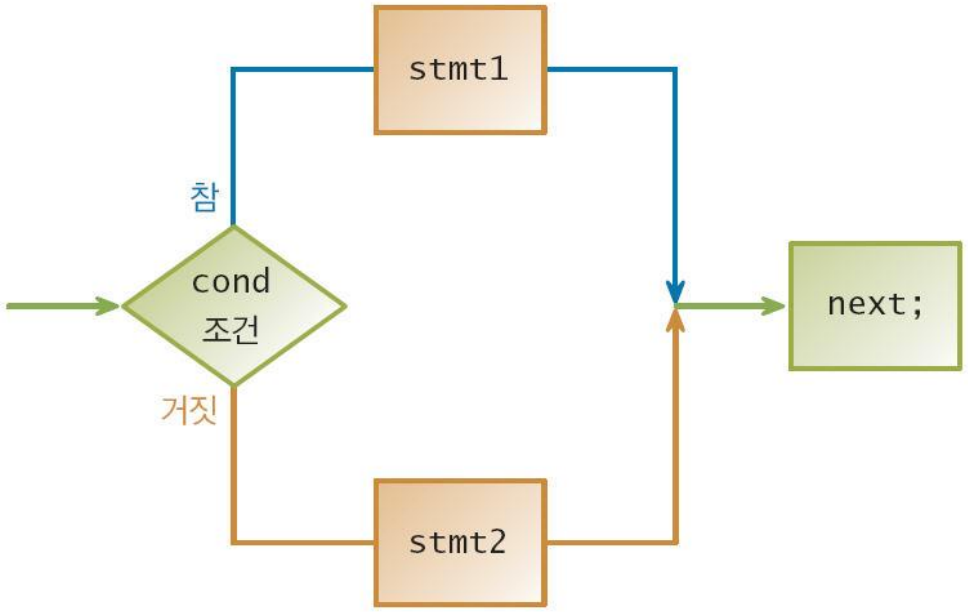
2. if 문

◆ 조건 만족 여부에 대한 선택

if else

❖ if else

- if 문은 조건이 만족되면 문장을 실행하는 구문
- 반대로 조건이 만족되지 않은 경우에 실행할 문장이 있다면 else를 사용



조건문 if else

```
if (cond)
        stmt1;
else
        stmt2;
next;
```

```
if (n % 2 == 0)
    printf("짝수");
else
    printf("홀수");

printf("입니다.\n");
```

[출처] 강환수 외, Perfect C 3판, 인피니티북스

실습예제 6-2 | Prj02 | 02evenifelse.c | 조건 if else로 짝수와 홀수 판정 | 난이도: ★

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main(void)
{
   int n;
   printf("정수 입력: ");
   scanf("%d", &n);

   if (n % 2) // if (n % 2 != 0)
   {
      printf("홀수\n");
   }
   else
   {
      printf("짝수\n");
   }

   //조건연산자 이용
   (n % 2) ? printf("홀수\n") : printf("짝수\n");

   return 0;
}
```

if else 문에서 조건식에 따른 실행문이 하나이더라도 블록을 구성해도 좋음

```c
if (n % 2 != 0)
   printf("홀수");
else
   printf("짝수");
```

else 앞의 블록에 세미콜론 ; 을 붙이면 문법 오류가 발생하니 주의가 필요하다.

결과

```
정수 입력: 5
홀수
홀수
```

```
정수 입력: 6
짝수
짝수
```

[출처] 강환수 외, Perfect C 3판, 인피니티북스

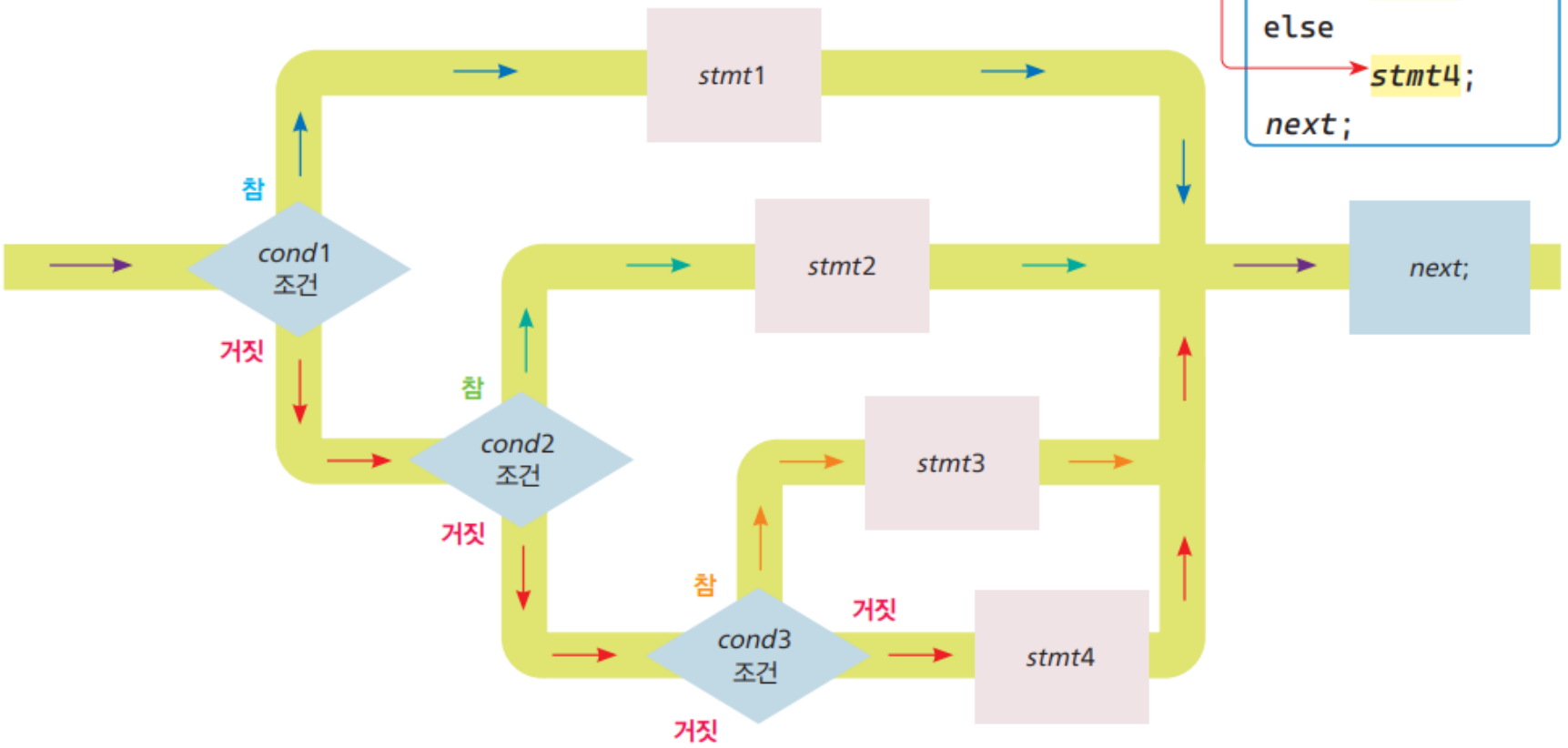
2. if 문

◆ 반복된 조건에 따른 선택 if else if

조건문 if else if

```
if (cond1)
        stmt1;
else if (cond2)
        stmt2;
else if (cond3)
        stmt3;
else
        stmt4;
next;
```

여러 개의 문장중 단 하나만 실행된다.



[출처] 강환수 외, Perfect C 3판, 인피니티북스

2. if 문

◆ 반복된 조건에 따른 선택 if else if

이 조건식은 첫 if의 조건식인 (point >= 90)이 만족되지 않고 체크되는 것이므로 결국 (!(point >= 90) && (point >= 80))이므로 80 이상에서 90 미만인 조건 (90>point && point>=80)이 만족된다.

```
if (point >= 90)
   printf("A\n");
else if (point >= 80)
   printf("B\n");
else if (point >= 70)
   printf("C\n");
else if (point >= 60)
   printf("D\n");
else
   printf("F\n");
```

필요하면 이와 같이 블록 사용이 가능하다.

```
if (point >= 90)
{
   printf("A\n");
}
else if (point >= 80)
{
   printf("B\n");
}
else if (point >= 70)
{
   printf("C\n");
}
else if (point >= 60)
{
   printf("D\n");
}
else
{
   printf("F\n");
}
```

[출처] 강환수 외, Perfect C 3판, 인피니티북스

2. if 문

실습예제 6-3 Prj03 03gradeif.c 조건 if else 문으로 평균평점에 따른 적정 구문 출력 난이도: ★

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03
04 int main(void)
05 {
06    double gpa;
07
08    printf("평균평점 입력: ");
09    scanf("%lf", &gpa);
10
11    if (gpa >= 4.3)
12       printf("최우등\n");
13    else if (gpa >= 3.8)
14       printf("우등\n");
15    else if (gpa >= 3.0)
16       printf("우수\n");
17    else
18       printf("3.0 미만\n");
19
20    return 0;
21 }
```

else if에 의해 if (gpa < 4.3 && gpa >= 3.8)의 의미가 됨

조건	단독 조건식	출력
평균평점 >= 4.3	gpa >= 4.3	최우등
4.3 > 평균평점 >= 3.8	gpa < 4.3 && gpa >= 3.8	우등
3.8 > 평균평점 >= 3.0	gpa < 3.8 && gpa >= 3.0	우수
3.0 > 평균평점	gpa < 3.0	3.0 미만

결과

```
평균평점 입력: 4.3
최우등
```

```
평균평점 입력: 3.9
우등
```

```
평균평점 입력: 3.3
우수
```

```
평균평점 입력: 2.7
3.0 미만
```

[출처] 강환수 외, Perfect C 3판, 인피니티북스

3. 중첩된 if 문

◆ if else if

```
if ( type == 1 )
{
    문장1;
}
else if ( type == 2 )
{
    문장2;
}
```

문장 1

```
if ( point >= 70 )
   printf("1종 면허 합격\n");
else
   printf("1종 면허 불합격\n");
```

문장 2

```
if ( point >= 60 )
   printf("2종 면허 합격\n");
else
   printf("2종 면허 불합격\n");
```

[출처] 강환수 외, Perfect C 3판, 인피니티북스

3. 중첩된 if 문

실습예제 6-4 | Prj04 | 04nestedif.c | 중첩된 if else 문으로 자동차 면허 합격 여부 판정 | 난이도: ★

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main(void)
{
   int type, point;

   printf("번호를 선택: 1(1종면허), 2(2종면허): ");
   scanf("%d", &type);
   printf("필기시험 점수 입력: ");
   scanf("%d", &point);

   if (type == 1)
   {
      if (point >= 70)
         printf("1종면허 합격\n");
      else
         printf("1종면허 불합격\n");
   }
   else if (type == 2)
   {
      if (point >= 60)
         printf("2종면허 합격\n");
      else
         printf("2종면허 불합격\n");
   }

   return 0;
}
```

결과

```
번호를 선택: 1(1종면허), 2(2종면허): 1
필기시험 점수 입력: 67
1종면허 불합격
```

```
번호를 선택: 1(1종면허), 2(2종면허): 1
필기시험 점수 입력: 77
1종면허 합격
```

```
번호를 선택: 1(1종면허), 2(2종면허): 2
필기시험 점수 입력: 58
2종면허 불합격
```

```
번호를 선택: 1(1종면허), 2(2종면허): 2
필기시험 점수 입력: 63
2종면허 합격
```

[출처] 강환수 외, Perfect C 3판, 인피니티북스

3. 중첩된 if 문

◆ 주의 사항

TIP [코딩 주의] 조건식에서 등호 연산자 ==를 대입 연산자 =으로 잘못 코딩하는 경우

위 소스에서 다음과 같이 조건식 (type == 1)을 실수로 (type = 1)로 잘못 코딩하면 (type = 1)의 결과는 대입 값인 1이므로 if 조건을 항상 실행하는 논리 오류가 발생하니 주의가 필요하다. 초보자들에게 자주 발생하는 오류이니 조심하자.

```
if (type = 1)
{
   ...
}
else if (type == 2)
{
   ...
```

TIP [코딩 주의] 조건식에서 등호 ==를 사용한 연산식에서 실수를 사용하는 경우의 문제

다음 코드의 결과는 '이상해요'가 출력된다. 변수 sum에는 float나 double의 문제로 실제 5.1이 아닌 5.1보다 조금 큰 실수가 저장된다. 그러므로 float나 double과 같은 실수를 관계 연산식이나 특히 등호 ==나 부등호 !=를 사용하는 경우 원하는 않는 결과가 발생할 수 있으니 가급적 사용하지 말자.

```
double a = 4.7, b = 0.4;
double sum = a + b;

if (sum == 5.1)
{
   printf("%s\n", "좋아요.");
}
else
{
   printf("%s\n", "이상해요.");
}
printf("%.20f ", sum);
```

```
이상해요.
5.10000000000000053291
```

[출처] 강환수 외, Perfect C 3판, 인피니티북스

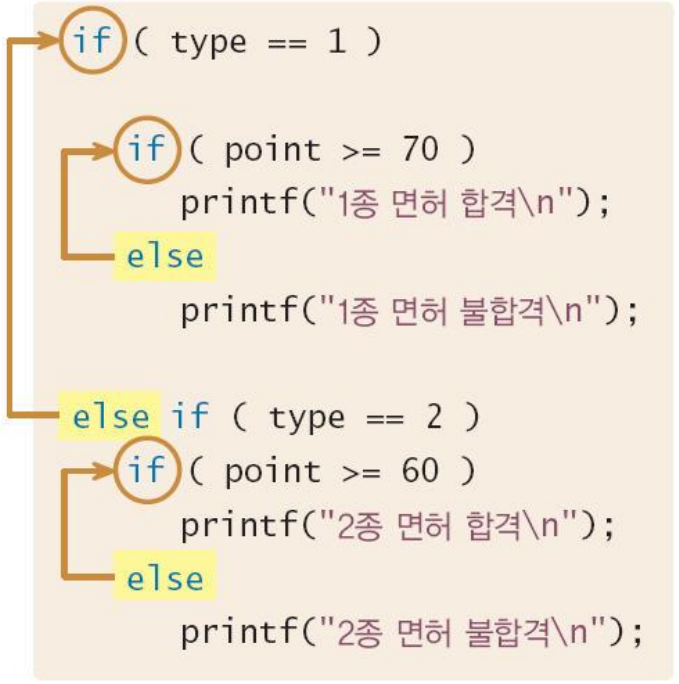
3. 중첩된 if 문

◆ 블록 표시와 else

❖ else의 혼란을 방지하려면 블록을 이용

- else는 문법적으로 같은 블록 내에서 else가 없는 가장 근접한 상위의 if 문에 소속된 else로 해석

```
if ( type == 1 )
{
   if ( point >= 70 )
      printf("1종 면허 합격\n");
   else
      printf("1종 면허 불합격\n");
}
   else if ( type == 2 )
{
   if ( point >= 60 )
      printf("2종 면허 합격\n");
   else
      printf("2종 면허 불합격\n");
```



[출처] 강환수 외, Perfect C 3판, 인피니티북스

4. 다양한 if 문의 이용과 조건연산자

조건 표현	형태	기준 변수	다양한 if 문으로 구성
온도가 32도 이상이면 폭염 주의를 출력	if	온도 temperature	if (temperature >= 32) printf("폭염 주의");
속도가 40km와 60km 사이이면 "적정 속도"라고 출력	if	속도 speed	if (40 <= speed && speed <= 60) printf("적정 속도");

운전면허 필기시험에서 60점 이상이면
합격, 아니면 불합격 출력
	if else	시험 성적
		point
			if (point >= 60)
			  printf("면허시험 합격");
			else
			  printf("면허시험 불합격");

구현 내용	조건연산자	if 문
두 수의 최대값 구하기	max = x > y ? x : y;	if (x > y) max = x; else max = y;
두 수의 최소값 구하기	min = x > y ? y : x;	if (x > y) min = y; else min = x;
절대값 구하기	abs = x >= 0 ? x : -x;	if (x >= 0) abs = x; else abs = -x;
홀수 짝수 구하기	a % 2 ? printf("홀수") : printf("짝수");	if (a % 2) printf("홀수"); else printf("짝수");

[출처] 강환수 외, Perfect C 3판, 인피니티북스

Ⅰ. 제어문과 조건문

1교시 수업을 마치겠습니다.

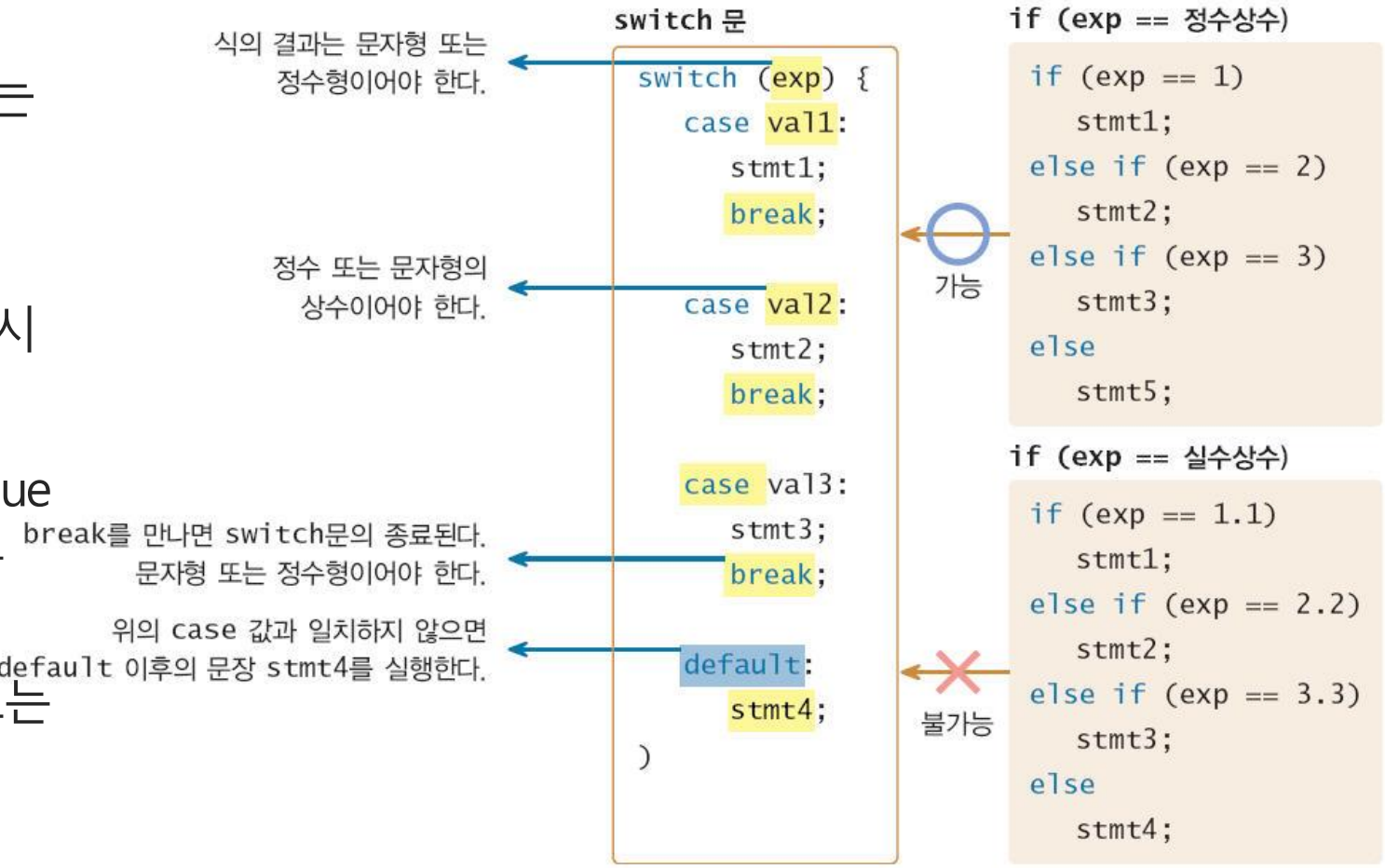
Ⅱ. 다양한 선택 switch 문

1. switch 문
2. break의 적절한 사용
3. 연산식 활용과 default 위치

1. switch 문

❖ 다양한 정수 또는 문자의 선택

- 연산식 exp의 결과값은 반드시 문자또는 정수
- case 다음의 value 값은 변수가 올 수 없으며 그 결과가 정수 또는 문자 상수
- default는 선택



[출처] 강환수 외, Perfect C 3판, 인피니티북스

1. switch 문

❖ break의 역할

- break 문이 없으면 break 문을 만나기 전까지 다음 case로 무조건 이동하여 내부 문장을 실행

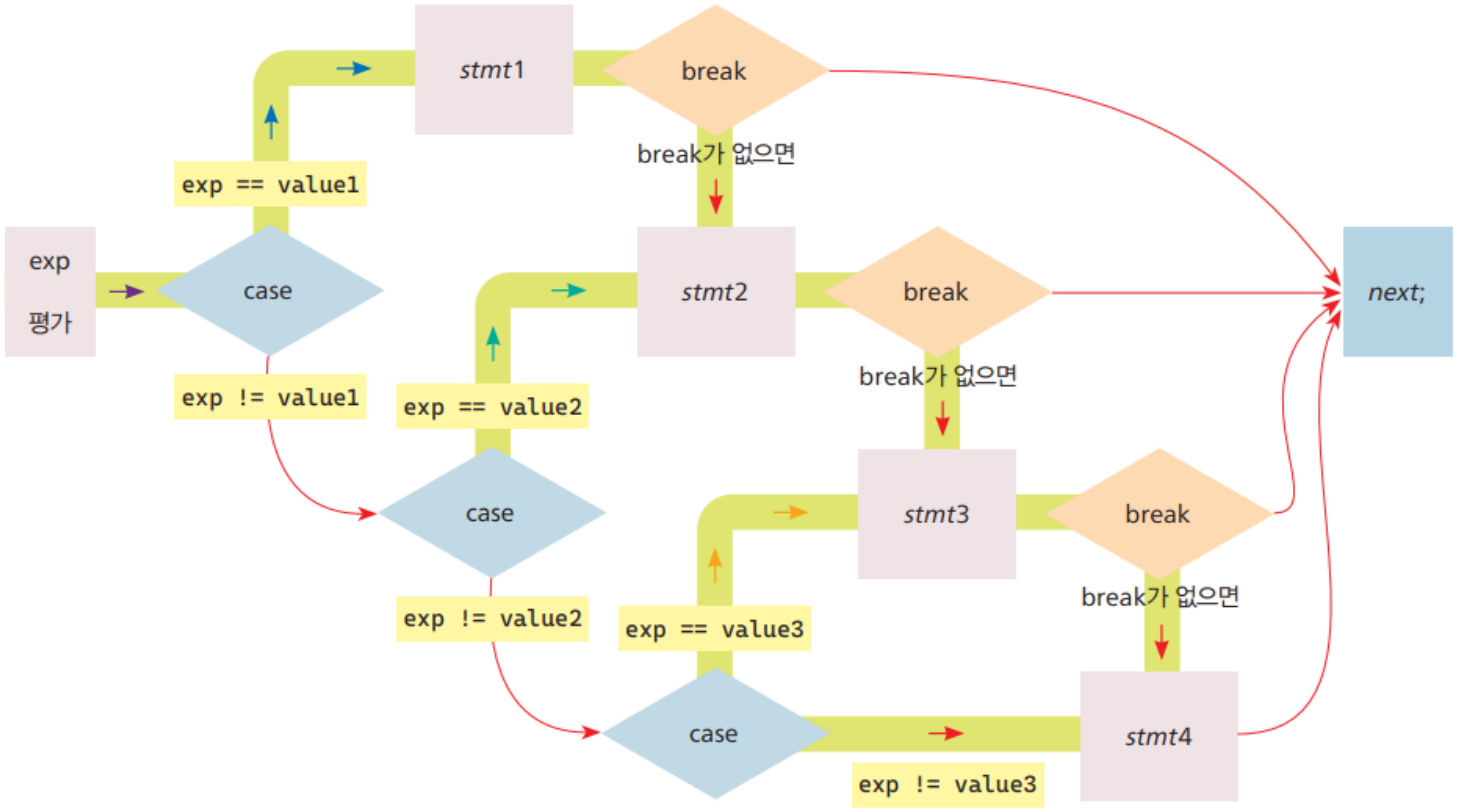


그림 6-20 switch 문의 제어흐름

[출처] 강환수 외, Perfect C 3판, 인피니티북스

1. switch 문

실습예제 6-6 | Prj06 | 06arithswitch.c | switch를 사용하여 두 실수의 사칙연산 수행 | 난이도: ★

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main(void)
{
   double x, y, result;
   int op;

   printf("두 실수 입력: ");
   scanf("%lf %lf", &x, &y);
   printf("연산종류 번호선택 1(+), 2(-), 3(*), 4(/): ");
   scanf("%d", &op);

   switch (op)
   {
      case 1:
         printf("%.2f + %.2f = %.2f\n", x, y, x + y);
         break;
      case 2:
         printf("%.2f - %.2f = %.2f\n", x, y, x - y);
         break;
      case 3:
         printf("%.2f * %.2f = %.2f\n", x, y, x * y);
         break;
      case 4:
         printf("%.2f / %.2f = %.2f\n", x, y, x / y);
         break;
      default:
         printf("번호를 잘못 선택했습니다.\n");
         break; //생략 가능
   }

   return 0;
}
```

만일 op가 2라면 case 2: 내부의 문장을 실행하고 종료된다.

case 1, 2, 3, 4 내부의 break 문은 반드시 필요하다.

결과

```
두 실수 입력: 3.765 6.987
연산종류 번호선택 1(+), 2(-), 3(*), 4(/): 1
3.77 + 6.99 = 10.75
```

```
두 실수 입력: 4.82 3.987
연산종류 번호선택 1(+), 2(-), 3(*), 4(/): 2
4.82 - 3.99 = 0.83
```

```
두 실수 입력: 3.986 4.826
연산종류 번호선택 1(+), 2(-), 3(*), 4(/): 3
3.99 * 4.83 = 19.24
```

```
두 실수 입력: 87.354 6.98
연산종류 번호선택 1(+), 2(-), 3(*), 4(/): 4
87.35 / 6.98 = 12.51
```

[출처] 강환수 외, Perfect C 3판, 인피니티북스

2. break의 적절한 사용

❖ case 이후의 상수

- switch 문에서 하나의 case에 여러 개의 정수를 콤마 나열 불가능

```
switch ( month )
{
   case 4 : case 5 :
      printf("%d월은 봄입니다.\n", month);
      break;

   case 6 : case 7 : case 8 :
      printf("%d월은 여름입니다.\n", month);
      break;

   ...

   default :
      printf("월(month)을 잘못 입력하셨습니다.\n");
}
```

```
case 4, 5 :      // 오류 발생
   ...
   break;

case 6, 7, 8 :   // 오류 발생
   ...
   break;
```

[출처] 강환수 외, Perfect C 3판, 인피니티북스

실습예제 6-7 | Prj07 | 07seasonswitch.c | 월에 따른 사계절 출력 | 난이도: ★

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main(void)
{
   int  month;
   printf("월(month)을 입력: ");
   scanf("%d", &month);

   switch (month)
   {
      case 4: case 5:
         printf("%d월은 봄입니다.\n", month);
         break;
      case 6: case 7: case 8:
         printf("%d월은 여름입니다.\n", month);
         break;
      case 9: case 10: case 11:
         printf("%d월은 가을입니다.\n", month);
         break;
      case 12: case 1: case 2: case 3:
         printf("%d월은 겨울입니다.\n", month);
         break;

      default:
         printf("월(month)을 잘못 입력했습니다.\n");
   }

   return 0;
}
```

month가 6, 7, 8이면 이 case로 들어와 여름이 출력된다.

결과

```
월(month)을 입력: 11
11월은 가을입니다.
월(month)을 입력: 5
5월은 봄입니다.
```

```
월(month)을 입력: 1
1월은 겨울입니다.
월(month)을 입력: 7
7월은 여름입니다.
```

[출처] 강환수 외, Perfect C 3판, 인피니티북스

3. 연산식 활용과 default 위치

실습예제 6-8 Prj08 08scoreswitch.c 점수에 따른 성적 부여

```c
#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>

int main(void)
{
   int  score;

   printf("점수 입력: ");
   scanf("%d", &score);

   switch (score / 10) {
   case 10: case 9:
      printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'A');
      break;
   case 8:
      printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'B');
      break;
   case 7:
      printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'C');
      break;
   case 6:
      printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'D');
      break;

   default:
      printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'F');
   }

   return 0;
}
```

정수의 나누기 결과는 정수라는 것을 활용한다.

100점도 있으므로 case 10이 필요하며, 90점 이상이고 100점 사이면 실행되는 case이다.

break를 생략하면 다음 case 내부 문장을 수행하므로 반드시 필요하다.

표 6-6 점수에 따른 성적처리를 위한 연산값

점수 예	점수 범위	(score / 10) 연산값	성적처리
100, 98, 95, 90	90 <= 점수 <= 100	9 또는 10	'A' 부여
80, 85, 88, 89	80 <= 점수 < 90	8	'B' 부여
80, 85, 88, 89	70 <= 점수 < 80	7	'C' 부여
80, 85, 88, 89	60 <= 점수 < 70	6	'D' 부여
30, 55, 58, 59	점수 < 60	그 외	'F' 부여

결과

```
점수 입력: 100
점수가 100 점으로 성적이 A 입니다.
```

```
점수 입력: 88
점수가 88 점으로 성적이 B 입니다.
```

```
점수 입력: 75
점수가 75 점으로 성적이 C 입니다.
```

```
점수 입력: 62
점수가 62 점으로 성적이 D 입니다.
```

[출처] 강환수 외, Perfect C 3판, 인피니티북스

3. 연산식 활용과 default 위치

❖ switch문 주의점

- 연산식 결과는 정수형 또는 문자형
- 각 case 뒤에 나오는 식은 상수식
  - 상수식에는 변수와 const 상수 사용 블가능
  - 리터럴 상수와 매크로 상수의 연산식은 사용 가능

❖ default

- 선택적으로 없거나 하나이며
- 어디에 위치해도 모든 case 처리를 하지 않은 경우 실행
- 다른 case가 뒤에 있다면 break가 필요

실습예제 6-9 Prj09 09scoreswitch2.c 잘못된 점수도 고려하여 점수에 따른 성적 부여 난이도: ★★

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main(void)
{
   int  score;
   printf("점수(0에서 100사이) 입력: ");
   scanf("%d", &score);
   if (score < 0 || score > 100)
   {
      printf("점수 입력이 잘못되었습니다.\n");
      return 0;
   }

   switch (score / 10)
   {
      default:
         printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'F');
         break;

      case 10: case 9:
         printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'A');
         break;
      case 8:
         printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'B');
         break;
      case 7:
         printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'C');
         break;
      case 6:
         printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'D');
         break;
   }

   return 0;
}
```

!(0 <= score && score <= 100)와 같으며, 점수가 음수이거나 100을 초과하면 조건을 만족한다.

결과

```
점수(0에서 100사이) 입력: 101
점수 입력이 잘못되었습니다.

점수(0에서 100사이) 입력: 94
점수가 94 점으로 성적이 A 입니다.

점수(0에서 100사이) 입력: 65
점수가 65 점으로 성적이 D 입니다.

점수(0에서 100사이) 입력: 55
점수가 55 점으로 성적이 F 입니다.
```

[출처] 강환수 외, Perfect C 3판, 인피니티북스

Ⅱ. 다양한 선택 switch 문

2교시 수업을 마치겠습니다.

Ⅲ. 반복문

1. while 문
2. do while 문
3. for 문

1. while 문

◆ while, do while, for 세 가지 종류의 반복 구문

조건이 참(0이 아닌 값)이어야 반복몸체를 실행

```
while ( <반복조건> )
{
   //반복몸체(loop body);
   <해야할 일>;
}
```

```
do
{
   //반복몸체(loop body);
   <해야할 일>;
} while ( <반복조건> );
```

조건이 참(0이 아닌 값)이어야 반복몸체를 다시 실행

```
for ( <초기화>; <반복조건>; <증감> )
{
   //반복몸체(loop body);
   <해야할 일>;
}
```

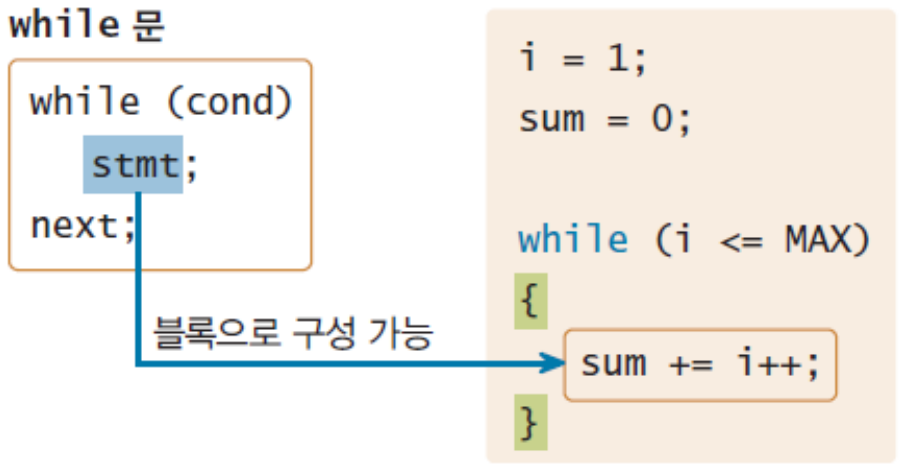
[출처] 강환수 외, Perfect C 3판, 인피니티북스

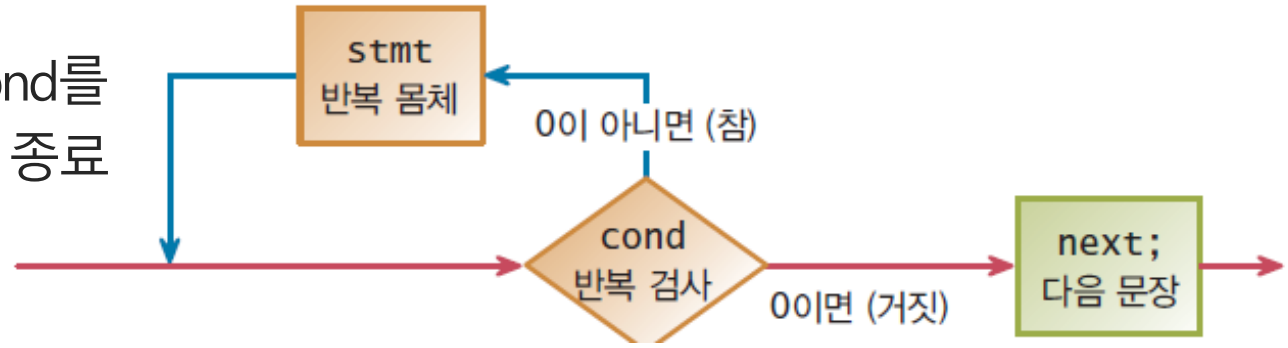
1. while 문

◆ while 문

❖ 문장 while (cond) stmt;

- 반복 조건인 cond를 평가하여 0(거짓)이면 while 문을 종료
- 0이 아니면(참) 반복 몸체인 stmt를 실행하고
- 다시 반복 조건 cond를 평가하여 while 문 종료 시까지 반복





[출처] 강환수 외, Perfect C 3판, 인피니티북스

1. while 문

◆ while 문

실습예제 7-3 | Prj03 | 03whilebasic.c | 특정 문자열을 여러 번 반복해 출력 | 난이도: ★

```
01  #include <stdio.h>
02
03  int main(void)
04  {
05      int count = 1;
06
07      while (count <= 3)
08      {
09          printf("반복, 재미있네요!\n");
10          count++;
11      };
12      printf("\n제어 변수 count => %d\n", count);
13
14      return 0;
15  }
```

while 반복몸체가 두 문장이므로 반드시 블록이 필요하며, 실수로 블록이 빠지면 논리 오류가 발생하여 무한 반복이 발생한다.

반복몸체로 제어변수 count를 1 증가시키는데, 결과값이 연산에 참여하지 않으므로 ++count도 가능하고, 결국 count += 1도 가능

while 문이 종료된 이후의 count 값은 3이 아니라 4라는 사실에 주의, 그러므로 출력 값은 4

결과

```
반복, 재미있네요!
반복, 재미있네요!
반복, 재미있네요!

제어 변수 count => 4
```

반복횟수	변수 count 값	조건식	조건식 평가	반목몸체
1	1	conut <= 3 1 <= 3	while (1)	printf("C 언어 재미있네요!\n"); count++;
2	2	conut <= 3 2 <= 3	while (1)	printf("C 언어 재미있네요!\n"); count++;
3	3	conut <= 3 3 <= 3	while (1)	printf("C 언어 재미있네요!\n"); count++;
4	4	conut <= 3 4 <= 3	while (0) while 종료	실행되지 못함

[출처] 강환수 외, Perfect C 3판, 인피니티북스

1. while 문

실습예제 7-4 | Prj04 | 04whilesum.c | while 반복으로 표준입력 실수를 모두 더하기 | 난이도: ★

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03
04 int main()
05 {
06    double number = 1, sum = 0;
07    while (number != 0.0)
08    {
09       printf("실수 입력 >> ");
10       scanf("%lf", &number);
11       sum += number;
12    }
13
14    printf("합 = %.2f\n", sum);
15
16    return 0;
17 }
```

초기 값은 1이지만 첫 while 조건을 통과하면서 새 표준입력으로 값이 대입됨

표준입력 값이 0이면 while 문을 종료

표준입력 값이 저장된 number를 계속 더하기

결과

```
실수 입력 >> 5.9
실수 입력 >> -3.4
실수 입력 >> 5
실수 입력 >> 0
합 = 7.50
```

논리 오류로 무한 반복 발생

```
int count = 1;
while (count <= 3)
   printf("C 언어 재미있네요!\n");
count++;
```

==

논리 오류로 무한 반복 발생

```
int count = 1;
while (count <= 3)
   printf("C 언어 재미있네요!\n");
   count++;
```

블록 처리로 원하는 구문 처리

```
int count = 1;
while (count <= 3)
{
   printf("C 언어 재미있네요!\n");
   count++;
}
```

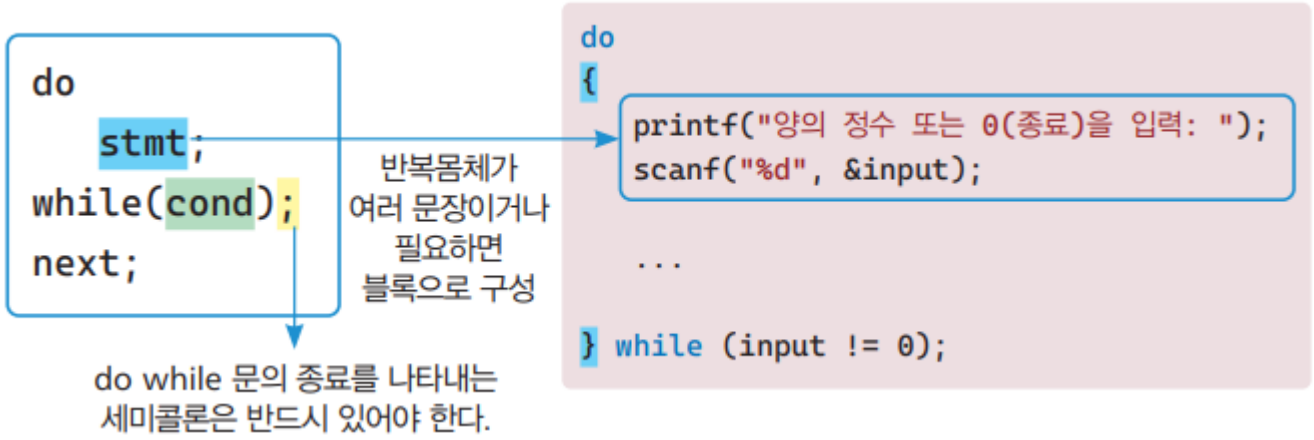
그림 7-12 while 블록의 중요성

[출처] 강환수 외, Perfect C 3판, 인피니티북스

2. do while 문

◆ do while 문

- do while 문은 반복 몸체 수행 후에 반복 조건을 검사
  - 특히 반복 횟수가 정해지지 않고 입력 받은 자료값에 따라 반복 수행의 여부를 결정하는 구문에 유용
- 반복 몸체에 특별한 구문이 없는 경우, do while 문의 몸체는 적어도 한 번은 실행
- do {...} while;과 같이 while 이후에 세미콜론 ;이 반드시 필요



[출처] 강환수 외, Perfect C 3판, 인피니티북스

2. do while 문

◆ do while 문

❖ 입력 후에 반복 검사를 진행하는 처리 과정

- 센티널 값(sentinel value): 반복의 종료를 알리는 특정한 자료 값

실습예제 7-5 | Prj05 | 05dowhile.c | 메뉴 주문 반복 | 난이도: ★

```
01  #define _CRT_SECURE_NO_WARNINGS
02  #include <stdio.h>
03
04  int main(void)
05  {
06     int input;
07     do
08     {
09        printf("[0]종료 [1]아메리카노 [2]카페라떼 [3]카푸치노 \n");
10        printf("주문할 커피 또는 종료(0)를 입력 >> ");
11        scanf("%d", &input);
12     } while (input != 0); //while (input);
13
14     return 0;
15  }
```

조건식 (input != 0)을 사용하므로 0이 아니어야 9번 줄로 이동하여 반복하며, 0이면 반복을 종료하고, 조건식 (input != 0)는 (input)과 같음

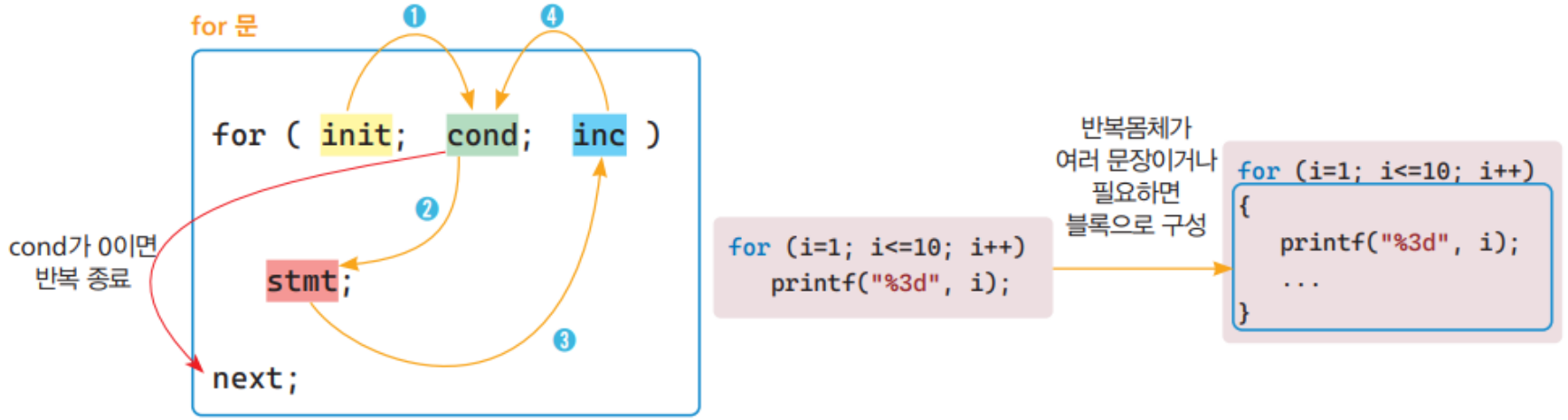
결과

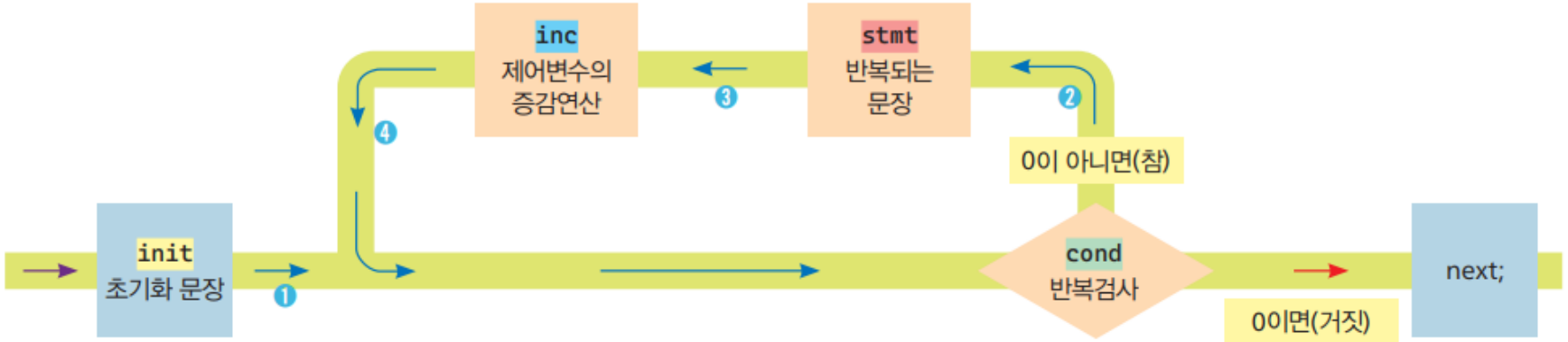
```
[0]종료 [1]아메리카노 [2]카페라떼 [3]카푸치노
주문할 커피 또는 종료(0)를 입력 >> 2
[0]종료 [1]아메리카노 [2]카페라떼 [3]카푸치노
주문할 커피 또는 종료(0)를 입력 >> 3
[0]종료 [1]아메리카노 [2]카페라떼 [3]카푸치노
주문할 커피 또는 종료(0)를 입력 >> 0
```

[출처] 강환수 외, Perfect C 3판, 인피니티북스

3. for 문

◆ for 문





[출처] 강환수 외, Perfect C 3판, 인피니티북스

3. for 문

◆ for 문

- 2개의 세미콜론은 반드시 필요
- 반복 조건 cond를 생략하면 반복이 계속

실습예제 7-6 | Prj06 | 06forbasic.c | for 구문으로 일정 횟수 반복 | 난이도: ★

```
01  #include <stdio.h>
02  #define MAX 5
03
04  int main(void)
05  {
06     int i;
07     for (i = 1; i <= MAX; i++)
08        printf("반복 %d\n", i);
09
10     printf("\nfor 종료 이후 i => %d\n", i);
11
12     return 0;
13  }
```

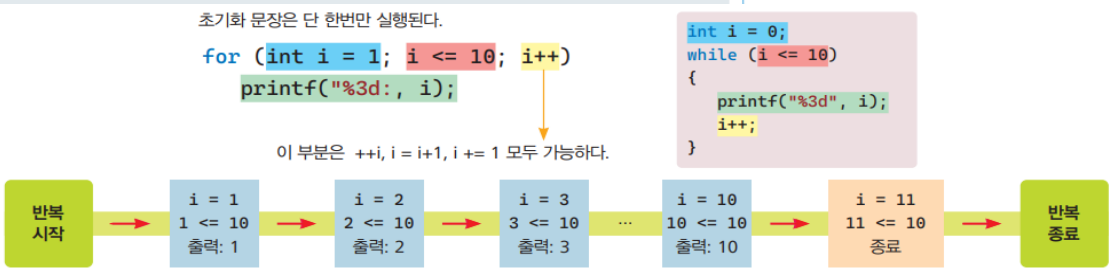
조건식 i <= MAX는 전처리 수행 후, MAX가 5로 대체되어 i <= 5 가 되며, i가 5보다 큰 6인 경우 조건식이 거짓이 되어 반복을 종료

증감의 i++ 는 반복몸체인 8번 줄의 문장이 실행된 이후 실행

결과

```
반복 1
반복 2
반복 3
반복 4
반복 5

for 종료 이후 i => 6
```



[출처] 강환수 외, Perfect C 3판, 인피니티북스

3. for 문

◆ 다양한 for 문

실습예제 7-7 | Prj07 | 07forcel2far3.c | for 문으로 3개의 섭씨 온도를 화씨 온도로 변환 | 난이도: ★

```
01  #include <stdio.h>
02  #define MAX 3
03  #define INCREMENT 10
04
05  int main(void)
06  {
07     double celsius = 12.46;
08
09     printf("  섭씨(C)   화씨(F)\n");
10     for (int i = 1; i <= MAX; i++, celsius += INCREMENT)
11     {
12        printf("%8.2f %8.2f\n", celsius, 9.0 / 5 * celsius + 32);
13     }
14
15     return 0;
16  }
```

매크로 상수 3을 정의, MAX는 반복 횟수 값으로 지정

for 문 초기화는 int i=1과 같이 변수 선언과 초기화도 가능, 변수 i는 for 문 내부에서만 사용 가능한 변수

이 부분은 여러 문장을 콤마로 나열이 가능하며, 제어변수도 1 증가시키고, 섭씨 온도도 증가분인 INCREMENT(10)만큼 증가시킴

결과

```
섭씨(C)   화씨(F)
12.46    54.43
22.46    72.43
32.46    90.43
```

[출처] 강환수 외, Perfect C 3판, 인피니티북스

3. for 문

TIP 반복 조건에서의 주의

반복 조건에서 등호나 부등호의 ==나 != 또는 대입연산자 =의 사용은 주의를 필요로 한다. 비교연산자 ==와 !=에서 피연산자로 실수는 가급적 사용하지 않도록 하자. 예를 한 가지 들자면, 다음과 같이 0.0에서 0.1씩 증가시켜 1.0까지 10회를 반복하고자 하는 경우, 반복 조건을 d != 1.0으로 하면 실수 연산의 오차로 인해 조건식 d != 1.0이 항상 참인 결과로 반복이 무한히 계속될 수 있다. 다음의 왼쪽 소스는 무한히 반복되나 오른쪽과 같이 d <= 1.0으로 조건을 검사하면 0, 0.1, 0.2, …, 1.0까지 출력된다.

```
for (double d = 0.0; d != 1.0; d += 0.1)
   printf("%f ", d);
```

```
for (double d = 0.0 d <= 1.0: d += 0.1)
   printf("%f ", d);
```

그림 7-19 실수의 연산에서 != 또는 ==의 문제

또한 대입연산자 =과 등호 연산 ==도 서로 혼동되지 않도록 유의하자. 다음 왼쪽 소스는 1 2 3 4 5가 출력되나 오른쪽 소스는 대입연산자인 =으로 잘못 사용하여 출력되는 것이 하나도 없다.

```
int i = 1;
while (!(i == 6))
{
   printf("%d ", i++);
}
```

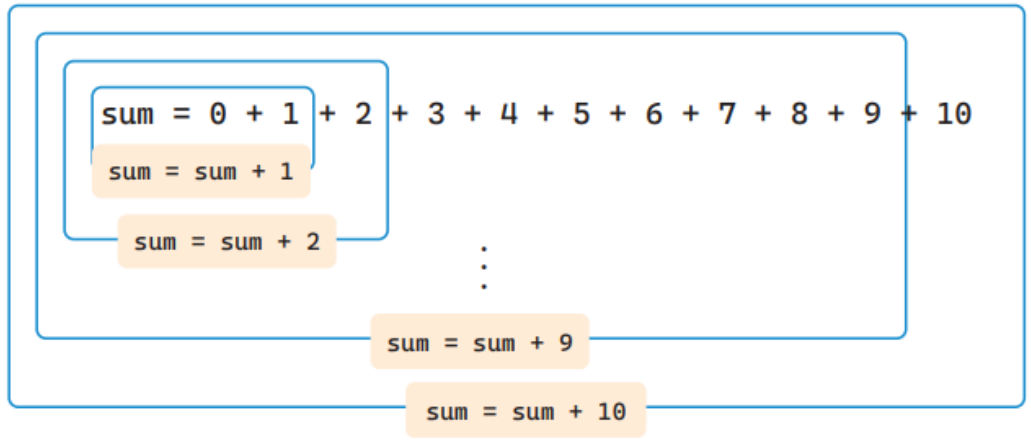
```
int i = 1;
while (!(i = 6))
{
   printf("%d ", i++);
}
```

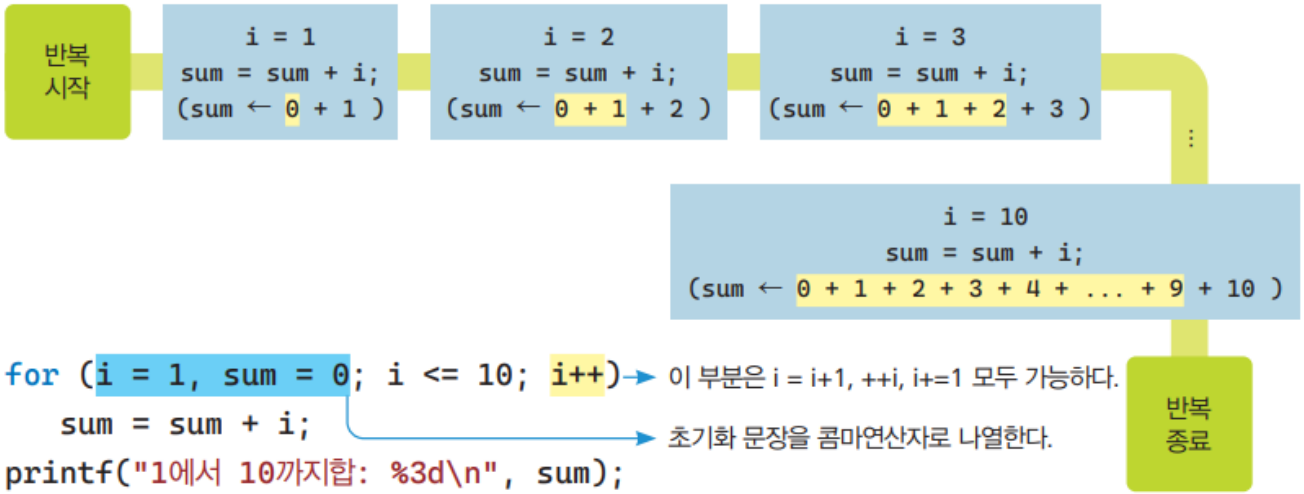
그림 7-20 ==를 =로 잘못 사용

[출처] 강환수 외, Perfect C 3판, 인피니티북스

3. for 문

◆ 합계를 구하기 위해 for 문 활용





[출처] 강환수 외, Perfect C 3판, 인피니티북스

3. for 문

◆ 합계를 구하는 다양한 for 문

실습예제 7-9 | Prj09 | 09forsum.c | 1에서 10까지의 합을 구하는 다양한 for 구문 | 난이도: ★

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05    int i, sum;
06    for (i = 1, sum = 0; i <= 10; i++) //++i도 가능
07       sum += i; // sum = sum + i;
08    printf("1 ~ 10 합: %d\n", sum);
09
10    for (i = 1, sum = 0; i <= 10; )
11       sum += i++;
12    printf("1 ~ 10 합: %d\n", sum);
13
14    for (i = 0, sum = 0; i <= 9; )
15       sum += ++i;
16    printf("1 ~ 10 합: %d\n", sum);
17
18    for (i = 1, sum = 0; i <= 10; sum += i++);   //반복몸체가 없는 for 문
19    printf("1 ~ 10 합: %d\n", sum);
20    for (i = 0, sum = 0; i <= 9; sum += ++i);    //반복몸체가 없는 for 문
21    printf("1 ~ 10 합: %d\n", sum);
22
23    return 0;
24 }
```

반복몸체인 sum += i는 들여쓰기가 반드시 필요하며, sum = sum + i의 축약

증감부분이 비어 있어도, 앞에 세미콜론은 반드시 필요

덧셈에 참여하는 값은 반복 조건을 통과한 값보다 1이 큰 정수이다.

for 문의 반복몸체는 따로 없으므로, 뒤에 for 문을 종료하는 세미콜론 ;이 반드시 필요하다.

결과

```
1 ~ 10 합: 55
1 ~ 10 합: 55
1 ~ 10 합: 55
1 ~ 10 합: 55
1 ~ 10 합: 55
```

[출처] 강환수 외, Perfect C 3판, 인피니티북스

3. for 문

◆ for 문

- ❖ 주로 반복 횟수를 제어하는 제어 변수를 사용
- ❖ 초기화와 증감 부분이 있는 반복문에 적합

◆ while 문

- ❖ 구조가 간단하므로 다양한 구문에 이용
- ❖ 반복 횟수가 정해지지 않고 특정한 조건에 따라 반복을 결정하는 구문에 적합
- ❖ for 문과 while 문은 서로 변환이 가능

[출처] 강환수 외, Perfect C 3판, 인피니티북스

3. for 문

```
pre
for ( A ; B ; C )
{
    body
}
```

변환가능

```
pre
A
while ( B )
{
    body
    C
}
```

```
sum = 0;
for (i = 1; i <= MAX; i++)
{
    sum += i;
}
```

변환가능

```
sum = 0;
i = 1;
while (i <= MAX)
{
    sum += i;
    i++;
}
```

sum += i++; 로 가능

[출처] 강환수 외, Perfect C 3판, 인피니티북스

3. for 문

실습예제 7-10 | Prj10 | 10inputsum.c | 1에서부터 표준입력한 양수까지의 합을 구하는 for와 while | 난이도: ★

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main(void)
{
   int i, sum, max;
   printf("양의 정수 입력 >> ");
   scanf("%d", &max);

   for (i = 1, sum = 0; i <= max; i++)    //++i도 가능
      sum += i; // sum = sum + i;
   printf("\nfor 문으로 구한 1에서 %d까지 합: %3d\n", max, sum);

   i = 1, sum = 0;
   while (i <= max)
   {
      sum += i;      // sum = sum + i;
      i++;           // ++i도 가능
   }
   printf("while 문으로 구한 1에서 %d까지 합: %3d\n", max, sum);

   return 0;
}
```

조건식 i <= max로, 증감은 i++로, 증감은 1만 증가시키면 되므로, ++i, i += 1, i = i + 1 도 가능

반복몸체의 두 번째 문장인 i++는 for 문에서 증감에 있던 문장으로, 증감은 1만 증가시키면 되므로, ++i, i += 1, i = i + 1 도 가능

결과

```
양의 정수 입력 >> 15

for 문으로 구한 1에서 15까지 합: 120
while 문으로 구한 1에서 15까지 합: 120
```

[출처] 강환수 외, Perfect C 3판, 인피니티북스

Ⅲ. 반복문

3교시 수업을 마치겠습니다.