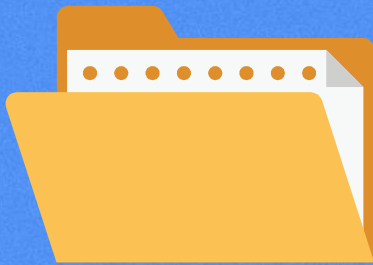




제06장

조건



01. 제어문 개요



제어문의 종류

- 순차적 실행

- 지금까지 배워 온 프로그램의 실행 순서의 원칙
 - 순차적(sequential) 실행

- 비순차적 실행의 제어문

- 프로그램의 실행 흐름을 제어하는 제어문 (control statement) 제공
 - 선택과 반복

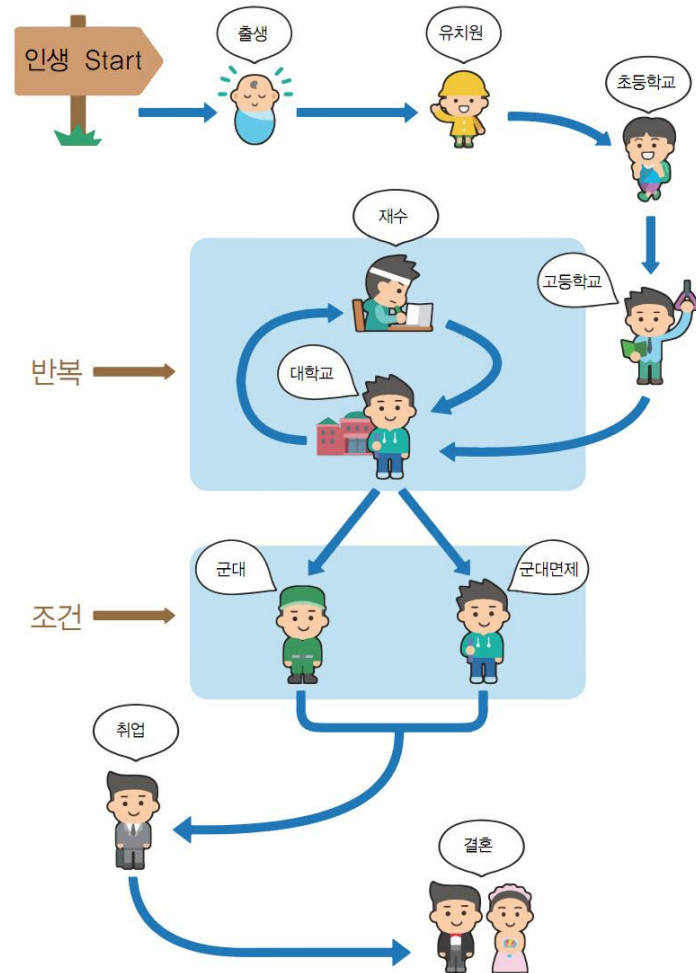


그림 6-1 인생여정과 프로그램의 실행 흐름

제어문 종류

- 조건선택

- 두 개 또는 여러 개 중에서 한 개를 선택하도록 지원하는 구문

조건선택

조건에 대한 선택 구문

- if
- if else
- if else if
- nested if
- switch

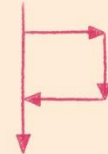


그림 6-2 조건선택

- 반복(순환)

- 정해진 횟수 또는 조건을 만족하면 정해진 몇 개의 문장을 여러 번 실행하는 구문

반복 순환

반복조건에 따라 일정영역의 반복 구문

- for
- while
- do while

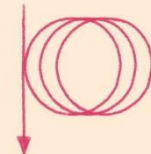
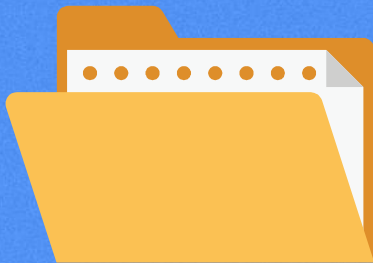


그림 6-3 반복순환



02. 조건에 따른 선택: if 문



if 문장

가장 간단한 if문의 형태

- if (cond) stmt;
 - 조건식 cond가 0이 아니면(참)
 - stmt를 실행
 - 0이면(거짓)
 - stmt를 실행하지 않음
 - 문장 stmt는 여러 문장이라면 블록으로 구성
 - if 문이 종료되면
 - 그 다음 문장 실행

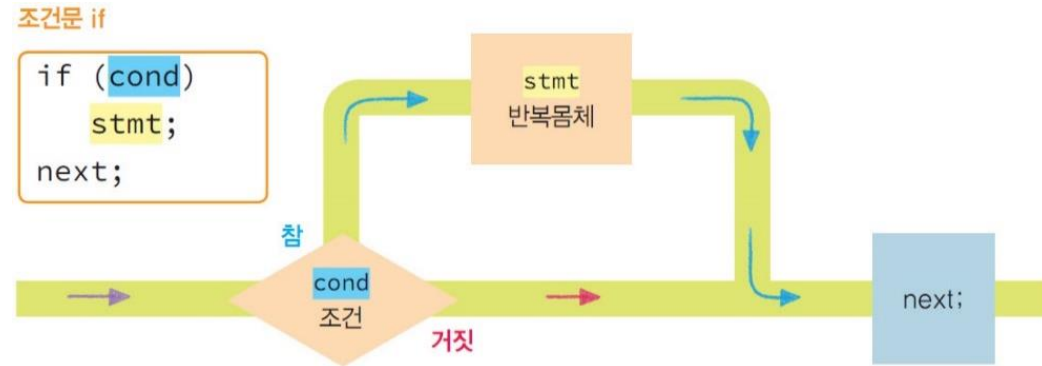


그림 6-7 if문과 두 개의 길

조건문 if

```
if (cond)
    stmt;
next;
```

cond가 만족되면 실행되는 문장

```
if (grade >= 3.2)
```

```
{
    printf("회사에 지원할 수 있습니다.\n");
}
```

```
printf("졸업을 축하합니다.\n");
```

블록이면 다음과 같이 블록 시작 {을 if열에 맞추고 조건 문장들을 들여쓰기 마지막에 행에 다시 블록 종료 }를 시작 열에 맞춘다.

그림 6-8 if 문

조건 if문

예제 if.c

- 입력하는 온도에 따라 다른 출력

주의사항

- if (cond); stmt;와 같이 조건식 다음에 세미콜론을 쓰는 경우
 - 문법오류가 없으며 if 문의 조건으로 인해 실행되는 문장은 하나도 없고, 이후 stmt는 무조건 실행

실습예제 6-1

if.c

```
01 // file: if.c
02 #define _CRT_SECURE_NO_WARNINGS
03
04 #include <stdio.h>
05
06 int main(void)
07 {
08     double temperature;
09
10     printf("현재 온도 입력: ");
11     scanf("%lf", &temperature);
12
13     if (temperature >= 32.0)
14     {
15         printf("폭염 주의보를 발령합니다.\n");
16         printf("건강에 유의하세요.\n");
17     }
18     printf("현재 온도는 섭씨 %.2f 입니다.\n", temperature);
19
20     return 0;
21 }
```

다음과 같이 블록 { 의 시작을 조건식 오른쪽으로 하기도 함

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

설명

14~17 조건식이 만족한 경우, 여러 문장을 실행하기 위해서는 잊지 말고 중괄호의 블록을 사용
15~16 조건식 (temperature >= 32.0)의 의미인 입력된 온도가 32도 보다 높으면 두 문장이 실행
18 현재 온도를 출력하는 문장은 항상 실행

실행결과

현재 온도 입력: 31.3
현재 온도는 섭씨 31.30 입니다.

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

학점이 3.2 미만이라도 다음 두 문장은 항상 실행되는 결과를 낳는다.

```
if (grade >= 3.2);  
    printf("회사에 지원할 수 있습니다.\n");  
  
printf("졸업을 축하합니다.\n");
```

if 문에서 조건식 grade >= 3.2에는 괄호가 필요하다. 문법오류로 인한 오류풍선이 발생하여 쉽게 알 수 있다.

```
if (grade >= 3.2)  
    printf("회사에 지원할 수 있습니다.\n");  
  
printf("졸업을 축하합니다.\n");
```

그림 6-9 if 문에서 세미콜론과 조건식 괄호에 주의

```

01 // file: if.c
02 #define _CRT_SECURE_NO_WARNINGS
03
04 #include <stdio.h>
05
06 int main(void)
07 {
08     double temperature;
09
10     printf("현재 온도 입력: ");
11     scanf("%lf", &temperature);
12
13     if (temperature >= 32.0)
14     {
15         printf("폭염 주의보를 발령합니다.\n");
16         printf("건강에 유의하세요.\n");
17     }
18     printf("현재 온도는 섭씨 %.2f 입니다.\n", temperature);
19
20     return 0;
21 }

```

다음과 같이 블록 { 의 시작을 조건식
오른쪽으로 하기도 함

```

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

```

설명

14~17 조건식이 만족한 경우, 여러 문장을 실행하기 위해서는 잊지 말고 중괄호의 블록을 사용
 15~16 조건식 (temperature >= 32.0)의 의미인 입력된 온도가 32도 보다 높으면 두 문장이 실행
 18 현재 온도를 출력하는 문장은 항상 실행

실행결과

현재 온도 입력: 31.3

현재 온도 입력: 34.678

if else 문장(1)

- 조건 만족 여부에 대한 선택 if else

- 조건문 if (cond) stmt1; else stmt2;
 - 조건 cond를 만족하면
 - stmt1을 실행
 - 조건 cond를 만족하지 않으면
 - stmt2를 실행
 - stmt1과 stmt2 둘 중의 하나를 선택하는 구문

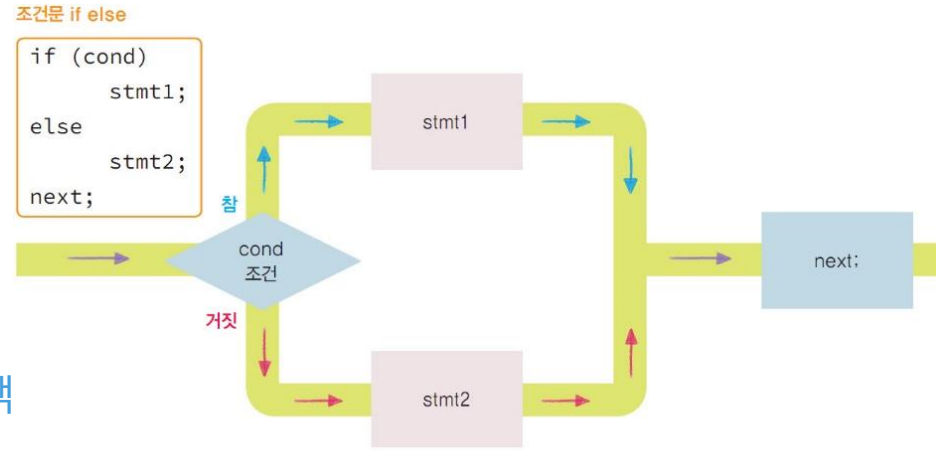


그림 6-10 if else문의 제어흐름

- 조건식: 나머지 연산자 %의 사용

- 조건식 (n % 2)
 - 0이 아니면(참) 홀수이고
 - 0이면(거짓) 짝수

조건문 if else

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

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

```
if (n % 2)
    printf("홀수");
else
    printf("짝수");
printf("입니다. \n");
```

그림 6-11 조건문 if else

if else 문장(2)

- 조건식에서 $(n \neq 0)$ 와 (n) 은 같은 식
 - 마찬가지로 $(n == 0)$ 와 $(!n)$ 도 같은 식

표 6-2 if 문에 사용하는 조건식

조건식	설명	예	
$(n \neq 0)$ (n)	n이 0이 아니어야 참인 연산식이므로 연산식 (n) 과 같음	<pre>if (n % 2 != 0) printf("홀수"); else printf("짝수");</pre>	<pre>if (n % 2) printf("홀수"); else printf("짝수");</pre>
$(n == 0)$ $(!n)$	n이 0이어야 참인 연산식이므로 연산식 $(!n)$ 과 같음	<pre>if (n % 2 == 0) printf("짝수"); else printf("홀수");</pre>	<pre>if (!(n % 2)) printf("짝수"); else printf("홀수");</pre>

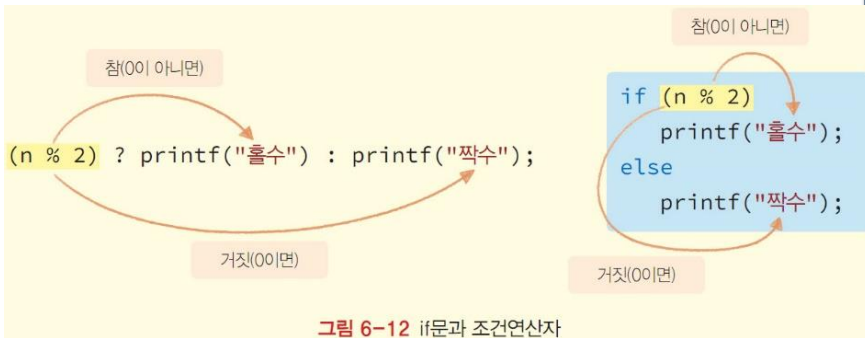
조건 if else 문

예제 ifelse.c

• 짝수와 홀수 구분

if와 조건연산자

- 조건 문장 if는 이미 배운 조건연산자와 매우 유사해서 좀 더 편리하게 활용



실습예제 6-2

ifelse.c

```
01 // file: ifelse.c
02 #define _CRT_SECURE_NO_WARNINGS
03
04 #include <stdio.h>
05
06 int main(void)
07 {
08     int n;
09
10     printf("정수 입력: ");
11     scanf("%d", &n);
12
13     if (n % 2) // if (n % 2 != 0)
14         printf("홀수");
15     else
16         printf("짝수");
17
18     printf("입니다.\n");
19
20     // 조건연산자 이용
21     (n % 2) ? printf("홀수") : printf("짝수");
22     printf("입니다.\n");
23
24     return 0;
25 }
```

설명

14 조건식을 만족하면 실행할 문장이 하나이므로 블록으로 하지 않았으나 블록으로 구성해도 좋음
21 if문과 동일한 기능의 조건연산자

실행결과

정수 입력: 5
홀수입니다.
홀수입니다.

정수 입력: 6
짝수입니다.
짝수입니다.

```

01 // file: ifelse.c
02 #define _CRT_SECURE_NO_WARNINGS
03
04 #include <stdio.h>
05
06 int main(void)
07 {
08     int n;
09
10     printf("정수 입력: ");
11     scanf("%d", &n);
12
13     if (n % 2) // if (n % 2 != 0)
14         printf("홀수");
15     else
16         printf("짝수");
17
18     printf("입니다.\n");
19
20     //조건연산자 이용
21     (n % 2) ? printf("홀수") : printf("짝수");
22     printf("입니다.\n");
23
24     return 0;
25 }

```

```

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

```

반복된 조건에 따른 선택 if else if

- 조건문 if else if 문

- 다음 문장에서

- stmt1에서 stmt4에 이르는 여러 문장 중에서 실행되는 문장은 단 하나!

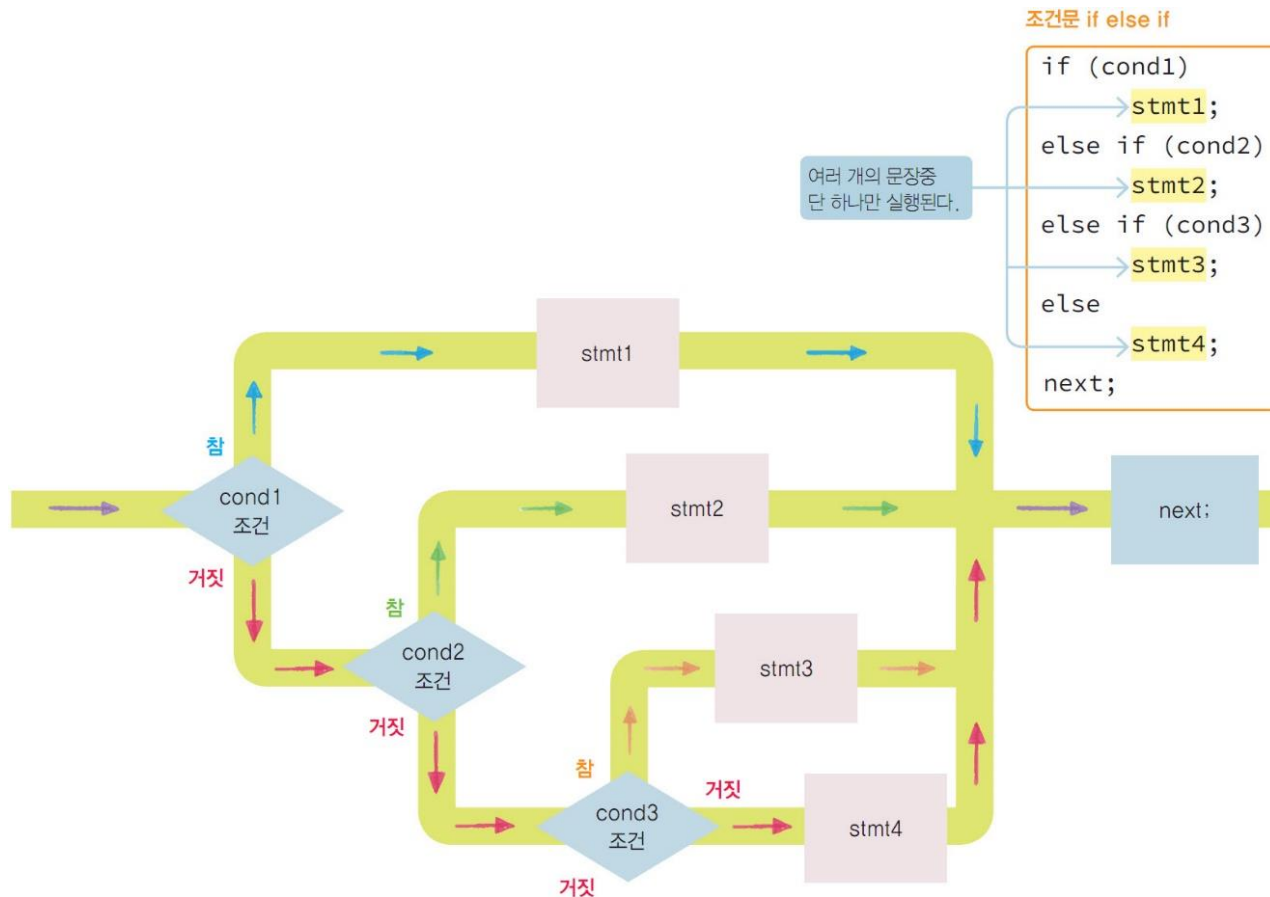


그림 6-13 조건문 if else if 문의 제어흐름

교과목 성적 처리

• 조건문 반복된 if else의 예

- 점수가 90이상이면 학점이 A
 - 80 이상에서 90 미만이면 B, 70 이상에서 80 미만이면 C
 - 60 이상에서 70 미만이면 D, 마지막으로 60 미만이면 F로 학점을 처리하는 모듈
- 가장 처음의 if는 if (point>=90)으로 시작
- 그 다음 if는 else 이후의 if이므로 else if (point>=80)으로 표현
 - 80 이상에서 90 미만인 조건(90>point && point>=80)이 만족

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");
}
```

그림 6-14 조건문 if else if else를 사용한 성적처리

예제 ifelseif.c

- 평균평점의 점수에 따라 출력이 달라지는 예제

실습예제 6-4

ifelseif.c

```
01 // file: ifelseifelse.c
02 #define _CRT_SECURE_NO_WARNINGS
03
04 #include <stdio.h>
05
06 int main(void)
07 {
08     double gpa;
09
10     printf("평균평점 입력: ");
11
12     scanf("%lf", &gpa);
13
14     if (gpa >= 4.3)
15         printf("성적이 최고 우수한 학생입니다.\n");
16     else if (gpa >= 3.8)
17         printf("성적이 매우 우수한 학생입니다.\n");
18     else if (gpa >= 3.0)
19         printf("성적이 우수한 학생입니다.\n");
20     else
21         printf("성적이 3.0 미만인 학생입니다.\n");
22
23     return 0;
24 }
```

설명

11 자료형 double이면 형식지정자는 %lf
21 else if (gpa >= 3.8) 자체가 if (gpa < 4.3 && gpa >= 3.8)의 의미
20 이 문장은 위 조건식이 모두 거짓인 경우 실행되는 문장

실행결과

평균평점 입력: 4.3

성적이 최고 우수한 학생입니다.

평균평점 입력: 3.9

성적이 매우 우수한 학생입니다.

평균평점 입력: 3.3

성적이 우수한 학생입니다.

평균평점 입력: 2.7

성적이 3.0 미만인 학생입니다.

표 6-3 평균평점에 따른 메시지 출력

조건	단독 조건식	출력
평균평점 >= 4.3	<code>gpa >= 4.3</code>	성적이 최고 우수한 학생입니다.
4.3 > 평균평점 >= 3.8	<code>gpa < 4.3 && gpa >= 3.8</code>	성적이 매우 우수한 학생입니다.
4.3 > 평균평점 >= 3.0	<code>gpa < 3.8 && gpa >= 3.0</code>	성적이 우수한 학생입니다.
3.0 > 평균평점	<code>gpa < 3.0</code>	성적이 3.0 미만인 학생입니다.

```
01 // file: ifelseifelse.c
02 #define _CRT_SECURE_NO_WARNINGS
03
04 #include <stdio.h>
05
06 int main(void)
07 {
08     double gpa;
09
10     printf("평균평점 입력: ");
11
12
13     scanf("%lf", &gpa);
14
15     if (gpa >= 4.3)
16         printf("성적이 최고 우수한 학생입니다.\n");
17     else if (gpa >= 3.8)
18         printf("성적이 매우 우수한 학생입니다.\n");
19     else if (gpa >= 3.0)
20         printf("성적이 우수한 학생입니다.\n");
21     else
22         printf("성적이 3.0 미만인 학생입니다.\n");
23
24     return 0;
25 }
```

중첩된 if 문

예제 nestedif.c

- if 문 내부에 if문이 존재하면 중첩된 if문

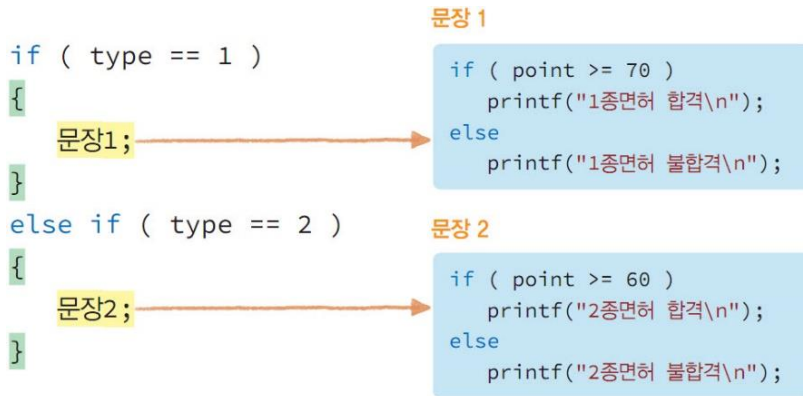


그림 6-15 중첩된 if

실습예제 6-5

nestedif.c

```
01 // file: nestedif.c
02 #define _CRT_SECURE_NO_WARNINGS
03
04 #include <stdio.h>
05
06 int main(void)
07 {
08     int type, point;
09
10     printf("번호를 선택: 1(1종면허), 2(2종면허): ");
11     scanf("%d", &type);
12     printf("필기시험 점수 입력: ");
13     scanf("%d", &point);
14
15     if (type == 1)
16     {
17         if (point >= 70)
18             printf("1종면허 합격\n");
19     }
20     else
21         printf("1종면허 불합격\n");
22     else if (type == 2)
23     {
24         if (point >= 60)
25             printf("2종면허 합격\n");
26         else
27             printf("2종면허 불합격\n");
28     }
29
30     return 0;
31 }
```

설명

11 변수 type에는 면허의 종류를 입력 저장
13 변수 point에는 면허 시험 성적을 입력 저장
15 외부 if 문
17~20 1종 면허인 경우, 점수를 합격여부를 판단하는 내부 조건문
24~27 2종 면허인 경우, 점수를 합격여부를 판단하는 내부 조건문

실행결과

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

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

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

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

블록 표시와 else

- 블록이 없는 경우 else 문장이 어느 if문의 소속인지 정확히 판단

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

- else의 혼란을 방지하려면 블록을 이용

- 예제

- else는 두 번째 if (age >= 65)의 else
- else는 문법적으로 문제가 없으나
 - 나이가 20 이상 65 미만도 미성년으로 구분되니 논리적으로 잘못
 - 조건식 (age >= 20) 이후에 블록을 반드시 사용한 후 else를 배치

```
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");
}
```

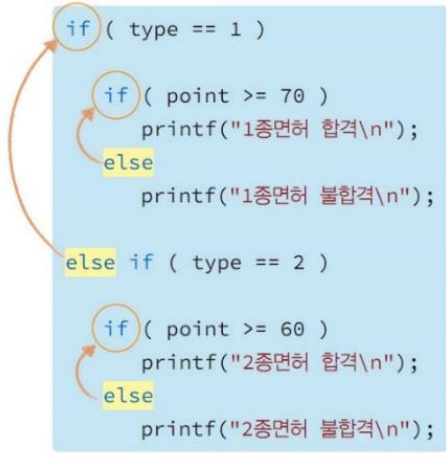
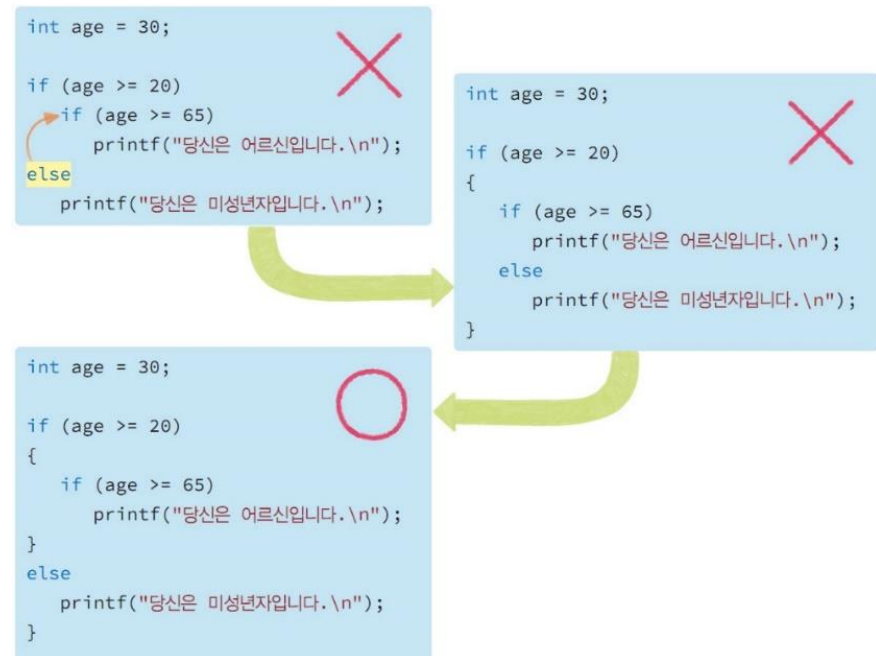


그림 6-16 if에서의 적절한 블록의 사용



```
int age = 30;

if (age >= 20)
    if (age >= 65)
        printf("당신은 어르신입니다.\n");
    else
        printf("당신은 미성년자입니다.\n");

int age = 30;

if (age >= 20)
{
    if (age >= 65)
        printf("당신은 어르신입니다.\n");
    else
        printf("당신은 미성년자입니다.\n");
}
```

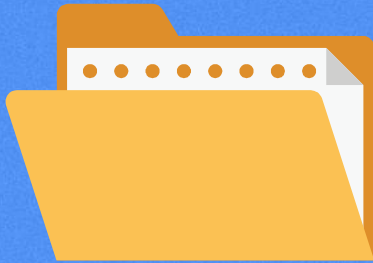
그림 6-17 else 구문의 주의

if와 조건연산자

- 조건연산자의 기능은 if문으로도 가능
 - 간단히 조건연산자로 구현할 수 있는 기능은 조건연산자를 사용하는 것이 더 효율적

표 6-5 조건연산자와 if

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



03. 다양한 선택 switch 문



switch 문장 개요

- 문장 if else가 여러 번 계속 반복되는 구문을 좀 더 간략하게 표현
- 주어진 연산식(=exp)의 값에 따라 case의 상수값과 일치하는 부분의 문장들을 수행

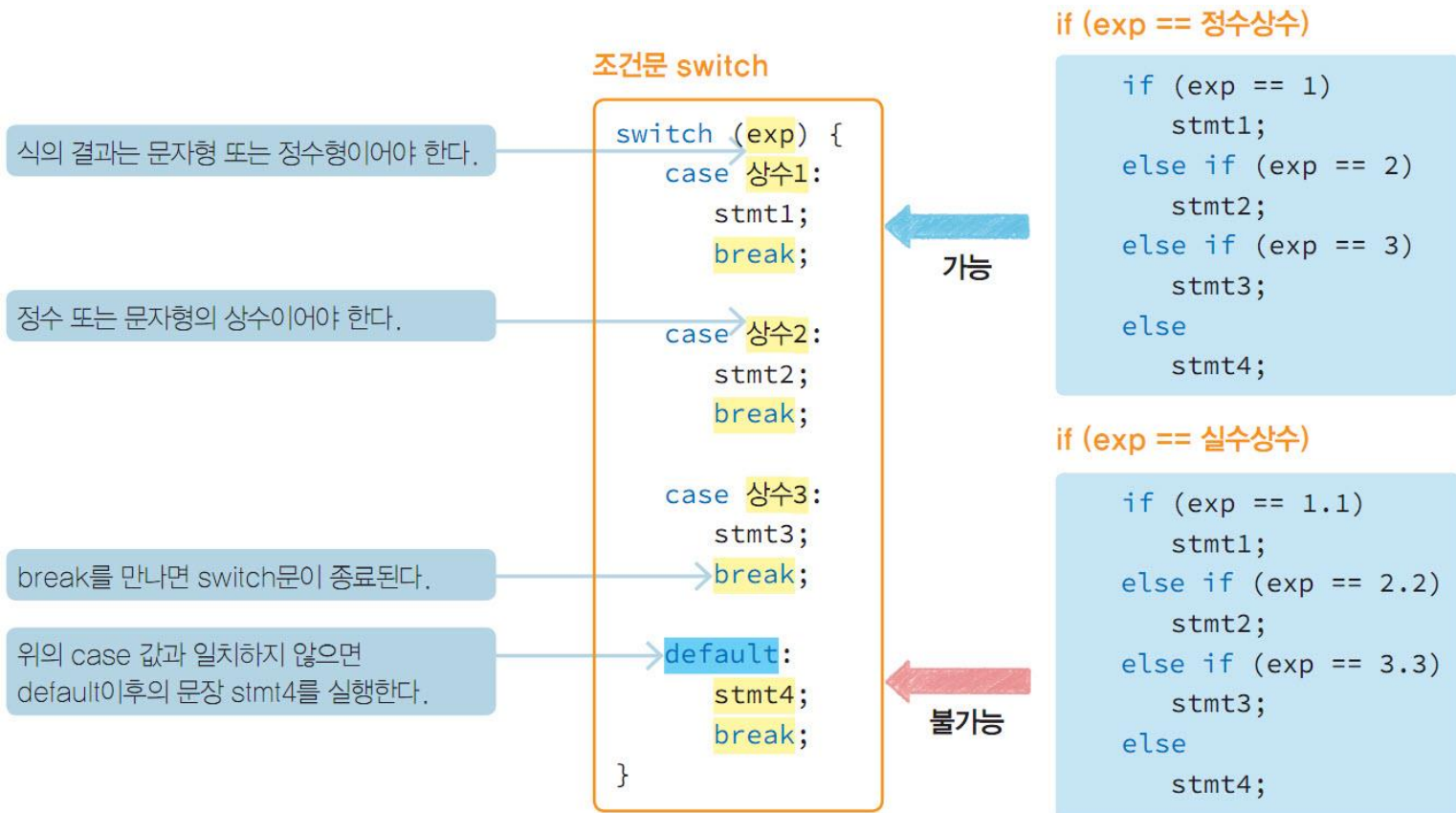


그림 6-19 switch 문 문장구조

switch 문

- **switch (exp) { ...} 문: switch, case, break, default는 키워드**
 - 표현식 exp 결과값: 반드시 정수 (또는 문자)
 - case의 값과 일치하는 항목의 문장 stmt1을 실행한 후 break를 만나 종료
 - case 다음의 value 값은 변수가 올 수 없으며
 - 상수식(constant expression)으로 그 결과가 정수 또는 문자 상수
 - break문이 없으면
 - break문을 만나기 전까지 다음 case의 내부로 이동하여 내부 문장을 실행

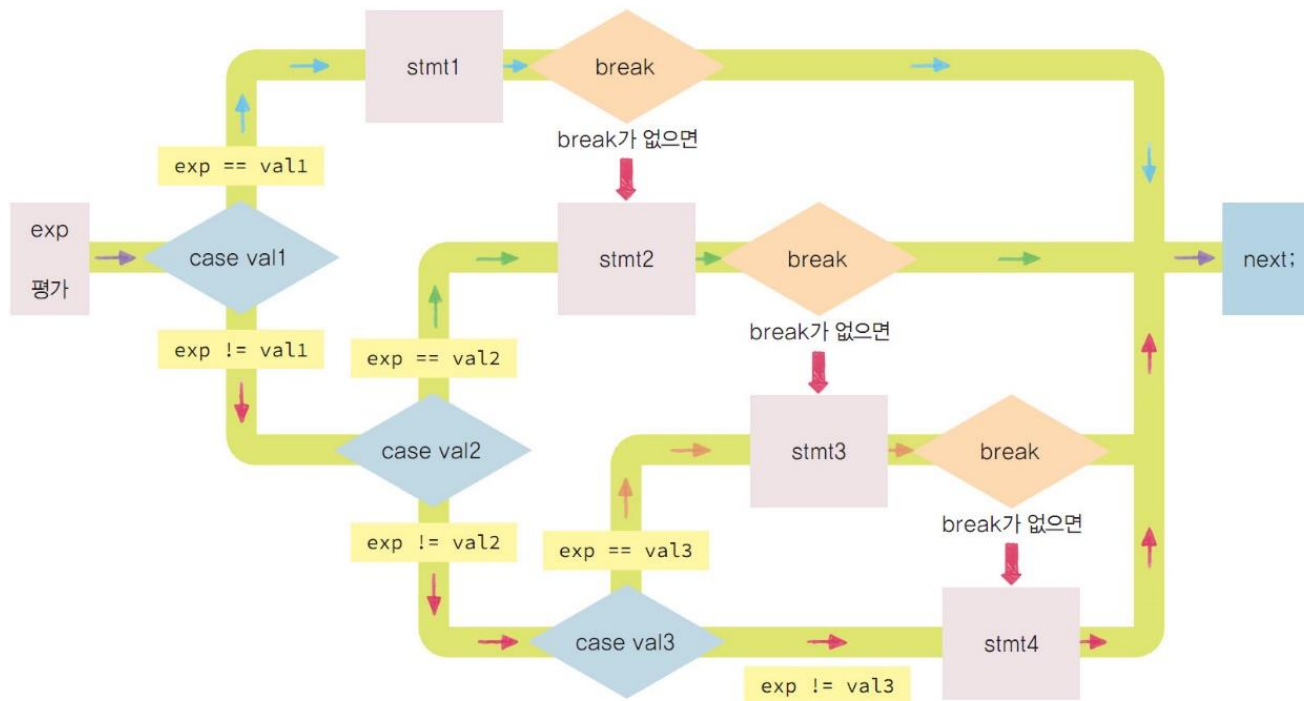


그림 6-20 switch 문의 제어흐름

switch 문

예제 switch.c

- 1에서 4까지 정수 중에서 선택한 번호에 따라 표준 입력한 두 실수의 더하기, 빼기, 곱하기 나누기를 실행하는 프로그램

28 break를 생략하면 논리오프로 바로 다음 default인 "번호를 잘못 선택했습니다."를 출력
32 입력한 연산종류가 1에서 4번이 아니라도 그 이후에 실행할 문장이 없으므로 break 생략 가능

실행결과

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

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

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

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

실습예제 6-7

switch.c

```
01 // file: switch.c
02 #define _CRT_SECURE_NO_WARNINGS //scanf() 오류를 방지하기 위한 상수 정의
03
04 #include <stdio.h>
05
06 int main(void)
07 {
08     double x, y, result;
09     int op;
10
11     printf("두 실수 입력: ");
12     scanf("%lf %lf", &x, &y);
13     printf("연산종류 번호선택 1(+), 2(-), 3(*), 4(/): ");
14     scanf("%d", &op);
15
16     switch (op) {
17     case 1:
18         printf("%.2f + %.2f = %.2f\n", x, y, x + y);
19         break;
20     case 2:
21         printf("%.2f - %.2f = %.2f\n", x, y, x - y);
22         break;
23     case 3:
24         printf("%.2f * %.2f = %.2f\n", x, y, x * y);
25         break;
26     case 4:
27         printf("%.2f / %.2f = %.2f\n", x, y, x / y);
28         break;
29
30     default:
31         printf("번호를 잘못 선택했습니다.\n");
32         break; //생략가능
33     }
34
35     return 0;
36 }
```

설명

12 scanf("두 실수 입력: %lf %lf", &x, &y); 이와 같이 쓰지 않도록 주의, 입력을 위한 프롬프트는 printf() 함수를 이용
19 break를 생략하면 논리오프로 바로 다음 case 2인 빼기도 수행
22 break를 생략하면 논리오프로 바로 다음 case 3인 더하기도 수행
25 break를 생략하면 논리오프로 바로 다음 case 4인 나누기도 수행


```
01 // file: switch.c
02 #define _CRT_SECURE_NO_WARNINGS //scanf() 오류를 방지하기 위한 상수 정의
03
04 #include <stdio.h>
05
06 int main(void)
07 {
08     double x, y, result;
09     int op;
10
11     printf("두 실수 입력: ");
12     scanf("%lf %lf", &x, &y);
13     printf("연산종류 번호선택 1(+), 2(-), 3(*), 4(/): ");
14     scanf("%d", &op);
15
16     switch (op) {
17     case 1:
18         printf("%.2f + %.2f = %.2f\n", x, y, x + y);
19         break;
20     case 2:
21         printf("%.2f - %.2f = %.2f\n", x, y, x - y);
22         break;
23     case 3:
24         printf("%.2f * %.2f = %.2f\n", x, y, x * y);
25         break;
26     case 4:
27         printf("%.2f / %.2f = %.2f\n", x, y, x / y);
28         break;
29
30     default:
31         printf("번호를 잘못 선택했습니다.\n");
32         break; //생략가능
33     }
34
35     return 0;
36 }
```

break의 적절한 사용

예제 seasonswitch.c

- 표준입력으로 1년 중 해당하는 월을 입력 받아 그 달에 맞는 계절을 출력하는 프로그램

OR 개념 처리

- case 문 내부에 break 문이 없다면
 - 일치하는 case 문을 실행하고
 - break 문을 만나기 전까지 다음 case 내부 문장을 실행
- case 4, 5와 같은 나열은 문법오류가 발생

```
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 : //오류발생
    ...
    break;

case 6, 7, 8 : //오류발생
    ...
    break;
```

그림 6-21 여러 개의 정수 선택에 따른 case 처리 방법

삼습예제 6-8

seasonswitch.c

```
01 // file: seasonswitch.c
02 #define _CRT_SECURE_NO_WARNINGS //scanf() 오류를 방지하기 위한 상수 정의
03
04 #include <stdio.h>
05
06 int main(void)
07 {
08     int month;
09
10     printf("년도의 월(month)을 입력: ");
11     scanf("%d", &month);
12
13     switch (month) {
14         case 4: case 5:
15             printf("%d월은 봄입니다.\n", month);
16             break;
17
18         case 6: case 7: case 8:
19             printf("%d월은 여름입니다.\n", month);
20             break;
21
22         case 9: case 10: case 11:
23             printf("%d월은 가을입니다.\n", month);
24             break;
25
26         case 12: case 1: case 2: case 3:
27             printf("%d월은 겨울입니다.\n", month);
28             break;
29
30         default:
31             printf("월(month)을 잘못 입력하셨습니다.\n");
32     }
33
34     return 0;
35 }
```

설명

```
12 scanf("%d", &month);에 의해 변수 month에 월이 입력
14 입력값이 4월, 5월이면
15 "~ 봄입니다." 출력
16 switch 문 종료
18 입력값이 6월, 7월, 8월이면
19 "~ 여름입니다." 출력
20 switch 문 종료
22 9월, 10월, 11월이면
23 "~ 가을입니다." 출력
24 switch 문 종료
26 12월, 1월, 2월, 3월이면
27 "~ 겨울입니다." 출력
```

```
01 // file: seasonswitch.c
02 #define _CRT_SECURE_NO_WARNINGS //scanf() 오류를 방지하기 위한 상수 정의
03
04 #include <stdio.h>
```

```
05
06 int main(void)
07 {
08     int month;
09
10     printf("년도의 월(month)을 입력: ");
11     scanf("%d", &month);
12
13     switch (month) {
14     case 4: case 5:
15         printf("%d월은 봄입니다.\n", month);
16         break;
17
18     case 6: case 7: case 8:
19         printf("%d월은 여름입니다.\n", month);
20         break;
21
22     case 9: case 10: case 11:
23         printf("%d월은 가을입니다.\n", month);
24         break;
25
26     case 12: case 1: case 2: case 3:
27         printf("%d월은 겨울입니다.\n", month);
28         break;
29
30     default:
31         printf("월(month)을 잘못 입력하셨습니다.\n");
32     }
33
34     return 0;
35 }
```