

학번: \_\_\_\_\_

성명: \_\_\_\_\_

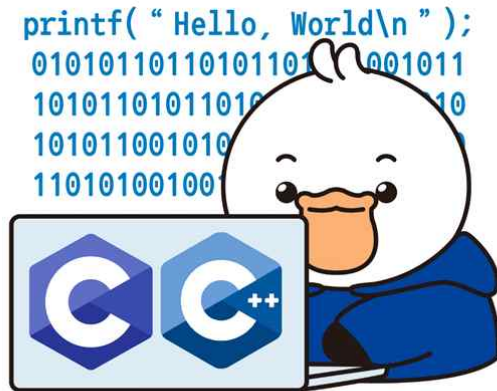
- 본 강의자료는 과학기술정보통신부 및 정보통신기획평가원에서 지원하는 『소프트웨어중심대학』 사업의 결과물입니다.
- 본 강의자료는 내용은 전재할 수 없으며, 인용할 때에는 반드시 과학기술정보통신부와 정보통신기획평가원의 '소프트웨어중심대학'의 결과물이라는 출처를 밝혀야 합니다.



대구가톨릭대학교  
컴퓨터소프트웨어학부  
School of Computer Software, Daegu Catholic University

## Part 4. 제어문 - 조건문

- 4.1 제어문
- 4.2 조건문 if
- 4.3 조건문 switch
- 4.4 Q&A
- 4.5 실습 및 과제
- 4.6 참고문헌



#### 4.1 제어문

- 제어문이란? 실행흐름 선택적
  - ✓ 프로그램의 을 변경하여 으로 프로그램을 실행할 수 있도록 하는 명령어
  - ✓ 지금까지 작성한 프로그램은 모두 하나의 방향으로 프로그램 실행
  - ✓ 이제부터 동일한 프로그램을 실행하더라도 실행 방향이 달라 다른 결과 출력 가능
  - ✓ 종류 - , ,

조건문, 반복문, 주판문(!)

#### 4.1 제어문

##### ■ 조건문이란?

- ✓ ☐에 따라 각기 다른 방향으로 프로그램을 수행하여 프로그램의 결과를 매번 다르게 출력하도록 할 수 있음
- ✓ 조건 선택문 또는 선택문
- ✓ 삼항 조건 연산자보다 복잡한 선택이 가능
- ✓ 종류
  - ☐ if문
    - 사용 방식에 따라 4가지로 분류 (if문 / if~else~문 / if~else if ~문 / 중첩된 if문)
  - ☐ switch문
    - 하나의 조건으로 여러가지 선택 수행 (switch case문)

제어된 조건

if문

switch문

#### 4.1 제어문

##### ■ 반복문이란?

- ✓ 동일한 문장 또는 블록을 ☐할 수 있도록 하는 명령문
- ex) 학생 100명의 성적을 처리하는 프로그램 작성 시
- ✓ 종류
  - ☐ / ☐ / ☐
- ✓ 각 반복문은 그 특성에 따라 각기 다른 방식으로 사용하지만, 일반적으로 상호 변환이 가능함.

반복적인 수행

while / do~while / for

#### 4.1 제어문

##### ■ 분기문이란?

- ✓ 프로그램의  할 때 사용
- ✓ 주로 조건문과 함께 사용됨
- ✓ 반복문 내부에서  위한 용도로 사용
- ✓ 종류

➤  /  /

continue, break, goto

흐름을 제어

반복문을 벗어나기

#### 4.2 조건문 if

##### ■ if문

- ✓ 다양한 조건문 중에서 가장 기본이라고 할 수 있음
- ✓ 영어 "if ~"는 "만약 ~ 라면" 으로 해석 → 일상 생활에서 흔히 접할 수 있는 상황을 처리하는 데 요긴하게 사용하는 문장
- ✓ 프로그램에 주어진 조건을 주고,  
조건을 만족했을 때 특정 명령을 수행할 수 있도록 할 수 있음

## 4.2 조건문 if - 조건문 예시

- [예시1] 사람의 나이가 만18세를 넘은 경우 투표권이 있다고 알려주는 시스템

- ✓ "만약" → if 로 변경
- ✓ 조건 → 조건 연산식으로 변경 후 괄호 안에 작성
- ✓ 조건에 부합할 때 실행해야 하는 문장 작성

만약 만18세 이상이라면  
투표권이 있다.

```
if (age >= 18)
    printf("투표권이 있다.\n");
```

<조건문 (if문) 예시 >

- 우리가 사용하는 자연어(한국어, 영어 등) 문장을 C 언어 문법에 맞게 수정하면, 프로그램 코드로 사용 가능

## 4.2 조건문 if - 조건문 예시

- [예시2] 자판기

- ✓ 동전을 넣고 원하는 음료수를 선택하여 구매할 수 있는 자동화 기계

- a. 동전이나 지폐 투입
- b. 투입 금액이 음료수 구매에 충분한 금액이 되면 음료수 선택 버튼 활성화
- c. 어떤 음료수가 선택되었는지에 따라 음료수를 배출하는 과정
- d. 음료수를 구매하고 남은 거스름돈 배출 과정

→ 모두 조건문으로 구현 가능

## 4.2 조건문 if - 조건문 예시

### ▪ [예시2] 자판기

if) 반환해야 하는 잔액이 670원일 때,

- a. 10원 짜리 동전 67개를 반환 (X)
- b. 500원 1개, 100원 1개, 50원 1개, 10원 2개 배출 (O)

→ 동전의 개수가 가장 적은 방식 선택

→ 이러한 동작도 조건문으로 구현 가능

## 4.2 조건문 if - 조건문 예시

### ▪ 자판기의 동작 순서

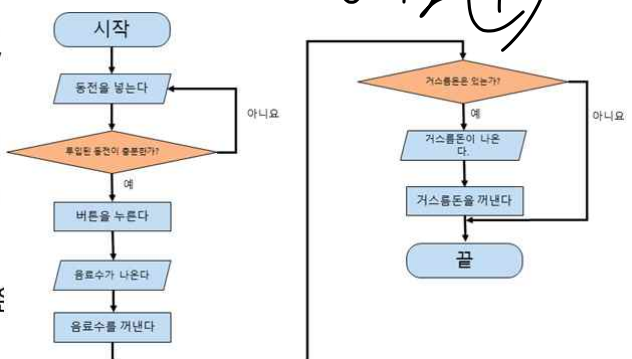
#### ✓ 조건문

➢ 주어진 조건에 따라   판별하여, 프로그램의 진행 흐름을 바꾸어 줌

➢   를 적절히 활용하여 조건에 맞는 조건식 생성

➢ 조건식이 명확하지 않은 경우, 프로그램은 프로그래머가 의도한 대로 동작 불가

참고 그림



< 자판기 동작 순서도 >

연상과 ( 관련 연상과 - 동리연상과 )

## 4.2 조건문 if

### ■ 조건에 따른 조건식 예시

- ✓ [예시] 놀이동산에서 만 2세부터 6세까지는 유아로 정의하고  
유아의 입장료를 5000원이라고 한다면,

➤  $(age \geq 2 \ \&\& \ age \leq 6)$  과 같은 조건식 필요

- 만약,  $\geq$  연산자가 아닌  $>$  연산자 사용 시

만 2세 유아들이 포함되지 않으므로 전혀 다른 결과를 가질 수도 있음

## 4.2 조건문 if

### ■ 조건에 따른 조건식 예시

- ✓ 프로그래밍을 처음 시작하는 사람들의 가장 흔한 실수 ①

➤ 수학의 조건식을 프로그래밍 조건식과 착각 하는 것

➤ ex)  $(2 \leq age \leq 6)$  는 에 의해  $((2 \leq age) \leq 6)$  로 동작

- 만약, 입장객의 나이가 20살이라고 할 경우

- $(2 \leq age)$ 의 조건식은 참이므로 대푯값은 1

- $(1 \leq 6)$ 의 조건식은 참이므로 대푯값은 1

➤ 결국, 만 20세의 입장객이 유아 요금을 지불하고 놀이동산에 입장하게 됨

*(2 ≤ age ≤ 6)*



## 4.2 조건문 if

### ■ 조건에 따른 조건식 예시

✓ 프로그래밍을 처음 시작하는 사람들의 가장 흔한 실수 ②

➢ 대입 연산자와 비교연산자를 혼돈하는 것

➢  와  는 서로 다른 연산자

➢ 조건식에는 특정 변수의 값이 상수값과 같은지 비교하는 경우에는 ==을 사용해야 함

대입연산자 = / 비교 연산자 ==

## 4.2 조건문 if

<조건에 따른 조건식의 예시>

조건	조건식	조건	조건식
2세 이상 6세 이하	(age >= 2 && age <= 6)	연봉이 4,000만원을 이하이고, 소유 자동차가 2000cc 미만인자 (단, 자녀가 2명 이상이거나 집을 소유하지 않은 경우 예외)	((salary <= 40000000 && car < 2000)    (children >= 2    house == 0))
90점 이상 100점 이하	(score >= 90 && score <= 100)	학점이 3.5이상이고, 수강신청 학점이 10학점 이상이라면 (단, 소득분위가 3분위 이하면 예외)	
80점 이상 90점 미만		비밀번호가 2020과 같다면	(pinNum == 2020)
잔액이 500원보다 크거나 같다면	(balance >= 500)	비밀번호가 2020과 같지 않다면	
숫자가 홀수라면	(num % 2)	상대 챔피언의 공격력이 내 챔피언의 방어력보다 작거나 같으면	(enemyAttack <= myDefence)
숫자가 5의 배수라면	(num % 5 == 0)	상대 챔피언의 체력이 (내 챔피언 공격력 - 상대 챔피언의 방어력) 보다 작다면	
숫자가 짝수라면		연도가 4의 배수 이면서 100의 배수가 아니거나, 400의 배수인 경우	
온도가 25도 이상이라면			
BMI가 18.5 이상 23 미만 (단, BMI는 체중의 제곱 곱하기 신장)	(kg*kg*m >= 18.5 && kg*kg*m < 23)		
BMI가 23 이상 25 미만 (단, BMI는 체중의 제곱 곱하기 신장)			



## 4.2 조건문 if

### if문

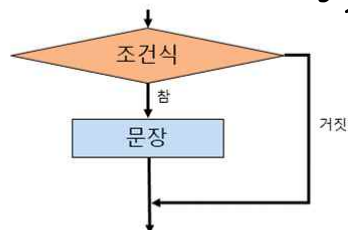
참이면 ( 0 이 아니라면 )

- ✓ 조건식이  다음에 오는 문장 또는 블록 실행
- ✓  다음의 문장 또는 블록을 실행하지 않고 다음으로 진행
- ✓ if문의 조건식의 참과 거짓
  - 조건식의 연산 결과가 0이면 거짓
  - 0이 아니면 참으로 간주

거짓이면 ( 0 이면 )

if(조건식) { 문장; }	if(조건식) 문장; 다음 문장;
-----------------------	--------------------------

< if문의 기본 문법 >



< if문의 순서도 >

## 4.2 조건문 if

### if문

- ✓ if 조건식으로 (score = 87)을 사용한다면, 연산식의 대푯값은 87이 되므로 참으로 간주
- ✓ 참인 경우, if문에 이웃한 문장 또는 블록 수행 후 "다음 문장;" 수행
- ✓ 거짓인 경우, 이웃한 문장을 실행하지 않고 "다음 문장;"으로 넘어 감

## 4.2 조건문 if

소스 4-1

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int num = -10;
5     if (num < 0) //관계연산자 조건식
6         printf("음수입니다.");
7     return 0;
8 }
```

출력 예 음수입니다.

## 4.2 조건문 if

소스 4-2

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int price = 500000;
5     int month;
6     printf("여행 희망 월을 입력하세요.");
7     scanf("%d", &month);
8     if (month == 7 || month == 8)
9         price = 900000;
10    printf("경비는 %d원입니다.", price);
11    return 0;
12 }
```

입력 예

8

출력 예

여행 희망 월을 입력하세요: 8  
경비는 900000원입니다.

## 4.2 조건문 if

소스 4-3

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int num;
5     if (!scanf("%d", &num))
6         printf("입력 실패");
7     return 0;
8 }
```

scanf() 함수의  을 조건식으로 사용

입력 예

d

출력 예

입력 실패

반환값 (리턴값)

## 4.2 조건문 if

소스 4-4

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int drinkPrice = 800, money;
5     printf("돈을 넣어주세요:");
6     scanf("%d", &money);
7     if (money >= drinkPrice) {
8         printf("음료수를 마실 수 있습니다.\n");
9         printf("가격은 %d원입니다.\n", drinkPrice);
10    }
11    return 0;
12 }
```

여러 개의 문장을 블록으로 묶어서 사용할 경우,  
블록은  로 묶어 사용

{ } 중괄호

입력 예

1000

출력 예

음료수를 마실 수 있습니다.  
가격은 800원입니다.

## 4.2 조건문 if

### ▪ if ~ else ~ 문

각각 다른 문장 실행

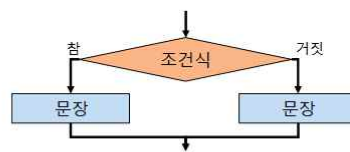
✓ 조건식이 참일 때와 거짓일 때  을 실행

만약 신장이 140Cm 이상이라면 탑승 가능 아니라면 탑승 불가	if (height >= 140) printf("탑승 가능\n"); else printf("탑승 불가\n");
--	--

< if ~ else ~ 문 예시 >

if(조건식) { 참 실행 문장; } else { 거짓 실행 문장; }	if(조건식) 참 실행 문장; else 거짓 실행 문장;
---	--

< if ~ else ~ 문의 기본 문법 >



< if ~ else ~ 문의 기본 문법 >

## 4.2 조건문 if

소스 4-5

```

1 /* 읽어 들인 정숫값 B는 A의 약수인가? */
2 #include <stdio.h>
3 /* 약수 판정 - A % B의 값이 0이면 약수 */
4 int main(void)
5 {
6     int a, b;
7     printf("2개의 정수를 입력하세요.");
8     scanf("%d %d", &a, &b);
9     if (a % b)
10        printf("%d는 %d의 약수가 아닙니다.\n", b, a);
11    else
12        printf("%d은(는) %d의 약수입니다.\n", b, a);
13    return 0;
14 }
  
```

입력 예

12 4

출력 예

2개의 정수를 입력하세요. 12 4  
4은(는) 12의 약수입니다.

## 4.2 조건문 if

소스 4-6

```

1 #include <stdio.h>
2 int main(void)
3 {
4     int a, b, difference;
5     printf("2개의 수를 입력하세요.");
6     scanf("%d %d", &a, &b);
7     if (a > b)
8         difference = a - b;
9     else
10        difference = b - a;
11    printf("두 값의 차이는 %d입니다. \n", difference);
12    return 0;
13 }

```

입력 예 12 10

출력 예 2개의 수를 입력하세요.  
두 값의 차이는 2입니다.

## 4.2 조건문 if

소스 4-6-1

```

1 #include <stdio.h>
2 int main(void)
3 {
4     int a, b, difference;
5     printf("2개의 수를 입력하세요.");
6     scanf("%d %d", &a, &b);
7     difference = a - b;
8     if (difference < 0)
9         difference = - difference;
10    printf("두 값의 차이는 %d입니다. \n", difference);
11    return 0;
12 }

```

입력 예 12 10

출력 예 2개의 수를 입력하세요.  
두 값의 차이는 2입니다.

## 4.2 조건문 if

소스 4-6-2

```

1 #include <stdio.h>
2 int main(void)
3 {
4     int a, b, difference;
5     printf("2개의 수를 입력하세요.");
6     scanf("%d %d", &a, &b);
7     difference = (a>b)?a-b:b-a;
8     printf("두 값의 차이는 %d입니다. \n", difference);
9     return 0;
10 }

```

삼항 연산자

if ~ else ~ 문은  와 상호 변환이 가능

입력 예 12 10

출력 예 2개의 수를 입력하세요.  
두 값의 차이는 2입니다.

## 4.2 조건문 if

소스 4-6-3

```

1 #include <stdio.h>
2 int main(void)
3 {
4     int a, b, difference;
5     printf("2개의 수를 입력하세요.");
6     scanf("%d %d", &a, &b);
7     difference = a - b;
8     difference = (difference>0)?difference:-difference;
9     printf("두 값의 차이는 %d입니다. \n", difference);
10    return 0;
11 }

```

삼항 연산자

if ~ else ~ 문은  와 상호 변환이 가능

입력 예 12 10

출력 예 2개의 수를 입력하세요.  
두 값의 차이는 2입니다.

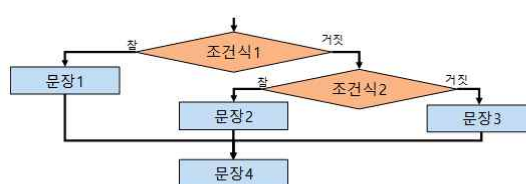
## 4.2 조건문 if

### ▪ if ~ else if ~ 문

- ✓ 하나의 조건에 대하여 참인 경우와 거짓인 경우를    하고자 할 때 사용
- ✓ 3개 이상으로 구분해서 수행해야 하는 프로그램을 작성할 때 유용하게 사용
- ✓ 첫 번째 if문의 조건식이 거짓인 경우,  
다시 else if의 조건식을 넣어 다시 참과 거짓을 판별하여 수행 가능

```
if(조건식1){
    문장1;
} else if (조건식2){
    문장2;
} else {
    문장3;
}
```

< if ~ else if ~ 문의 기본 문법 >



< if ~ else if ~ 문의 순서도 >

## 4.2 조건문 if

소스 4-7

```
1 #include <stdio.h>
2 int main(void){
3     int age, fee;
4     scanf("%d", &age);
5     if (age >= 18) {
6         fee = 10000;
7     }
8     else if (age >= 13) {
9         fee = 8000;
10    }
11    else
12    {
13        fee = 5000;
14    }
15    printf("요금은 %d원입니다.", fee);
16    return 0;
17 }
```

- ✓ 여러 구간으로 나누어져 있는 조건을 처리해야 하는 경우  
if ~ else if ~ 문을 사용함

입력 예	15
출력 예	요금은 8000원입니다.



## 4.2 조건문 if

소스 4-8

```

1 #include <stdio.h>
2 int main(void) {
3     int score = 0;
4     printf("C프로그래밍 점수 입력: ");
5     scanf("%d", &score);
6     if (score >= 90) {
7         printf("당신의 학점은 A입니다.\n");
8     }
9     else if (score >= 80) {
10        printf("당신의 학점은 B입니다.\n");
11    }
12    else if (score >= 70) {
13        printf("당신의 학점은 C입니다.\n");
14    }
15    else if (score >= 60) {
16        printf("당신의 학점은 D입니다.\n");
17    }
18    else {
19        printf("당신의 학점은 F입니다.\n");
20    }
21    return 0;
22 }

```

✓ 연속적인 구간에서 특정 값이

결정하

는데 매우 유용

예는 구간에 속하냐

입력 예	90
출력 예	C프로그래밍 점수 입력 : 90 당신의 학점은 A입니다.

## 4.2 조건문 if

### ■ 중첩(nested) if 문

✓ 프로그램으로 구현해야 하는 조건이 복잡한 경우

→ 복합적인 조건식 or 여러 개의 if문을 사용하여 구현

✓ 하나의 if문 안에 또 다른 if문을 사용하는 것

## 4.2 조건문 if

소스 4-9

```

1 #include <stdio.h>
2 int main(void)
3 {
4     int age, fee = 0;
5     char localPeople;
6     scanf("%d %c", &age, &localPeople);
7     if (age >= 18) {
8         if (age >= 65) fee = 5000;
9         else fee = 10000;
10    } else {
11        if (age >= 13) fee = 8000;
12        else if (age >= 7) fee = 6000;
13        else fee = 3000;
14    }
15    if (localPeople == 'Y') fee -= 1000;
16    printf("요금은 %d원입니다.", fee);
17    return 0;
18 }

```

입력 예	15 Y
출력 예	요금은 7000원입니다.

## 4.2 조건문 if

### ■ 다수의 if문 및 중첩 if문의 사용 시, if ~ else ~ 사용의 주의점

- ✓ else가 어떤 if에 속하는지를 명확히 하지 않아 문제 발생
  - *lacking else 문제*
- ✓ 두 개 이상의 if문에 하나의 else가 사용된 경우, else가 어떤 if와 짝을 이루어 동작하는지 명확히 해야 함 *중괄호*
  - 보통  를 이용하여 각 if문의 범위를 명확히 하면 대부분 해결 가능
- ✓ 프로그램을 작성할 때 잘못된 들여쓰기로 인하여 소스코드를 읽는 사람으로 하여금 명확하게 이해하지 못하는 경우도 종종 발생

## 4.2 조건문 if

< dangling else 문제 예시 >

A	B	C
<pre>#include &lt;stdio.h&gt; int main() {     int age;     scanf("%d", &amp;age);     if(age &gt;= 18)         if(age &gt;= 65)             printf("그룹1\n");         else             printf("그룹2\n");     return 0; }</pre>	<pre>#include &lt;stdio.h&gt; int main() {     int age;     scanf("%d", &amp;age);     if(age &gt;= 18)         if(age &gt;= 65)             printf("그룹1\n");         else             printf("그룹2\n");     return 0; }</pre>	<pre>#include &lt;stdio.h&gt; int main() {     int age;     scanf("%d", &amp;age);     if(age &gt;= 18) {         if(age &gt;= 65)             printf("그룹1\n");     }     else         printf("그룹2\n");     return 0; }</pre>

## 4.3 조건문 switch

### ▪ switch 문

- ✓ switch ~ case ~문 이라고도 함 *하나의 정수형 변수*
- ✓                     을 이용하여 여러 케이스 중 *값* 선택적으로 프로그램을 수행할 수 있도록 함
- ✓ 정수형 변수 값을 이용하여 여러 경우의 수에 따라 프로그램의 흐름 변경 가능
  - 정수형 값을 이용하는데,           을 포함
  - if문의 경우, 조건식의 참과 거짓에 의하여 프로그램 흐름 변경

```
switch(정수식){
case 정숫값1 :
    문장1;
    break;
case 정숫값2 :
    문장2;
    break;
default :
    문장n;
    break;
}
```

< switch문의 형식 >

*문자형 값*

#### 4.3 조건문 switch

##### ▪ switch 문

✓ switch 다음의 괄호 안에는 정수형 변수 또는 정수를 대푯값을 가지는 수식

➢ 정수형 값을 가질 수 있으면 사용 가능하므로, 도 사용 가능

✓ switch 문으로 수행할 문장들은 중괄호 내부에 작성

➢ 와 같은 키워드 함께 사용

✓ case의 정수값과 정수식의 결과가 같은 경우, 다음의 문장 수행

➢ 정수식의 값이 같은 case의 문장을 만나면 이후의 모든 문장 수행

➢ 이 경우 다른 case의 문장들도 수행하게 되는데,

이를 방지하고자  키워드 이용하여 switch문 벗어나게 함

#### 4.3 조건문 switch

##### ▪ switch 문

✓ break는 현재 자신을 둘러싼 블록을 벗어날 수 있도록 하는 분기문

➢ switch문에서 다른 case의 문장을 수행하지 않기 위해 빠져나가는 기능

✓ case 다음에는 항상 정수값(문자 포함)을 적고 를 적어야 함

✓ 는 다른 모든 case에 해당하지 않는 경우에 수행하는 부분

➢ 정수식의 값이 case의 값과 대응하지 않는 경우 default 다음의 문장 수행

➢ 위치는 상관 없으나, 일반적으로 가장 마지막에 작성

### 4.3 조건문 switch

#### ▪ switch 문과 if ~ else if ~문

- ✓ if ~ else if ~문으로 표현한 코드는 대개의 경우 switch문으로도 표현 가능
  - 물론, 반대의 경우도 가능
  - 프로그래머의 취향에 따라 선택 가능
- ✓ switch문의 경우 모든 경우에 대한 동일한 실행 시간 가짐
  - 반면, if ~ else if ~의 경우 조건 비교의 횟수가 각 상황마다 다르기 때문에 동일한 시간을 보장하기 어려움
- ✓ 모든 선택에 대한 동일한 수행 시간을 보장해야 하는 경우, switch문 권장
  - 특히, 게임의 입력에 따른 동작을 선택해야 하는 경우 switch문 사용

### 4.3 조건문 switch

소스 4-10

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int select;
5     printf("메뉴 선택 (1:사과주스 2:딸기주스 3:키위주스 4:오렌지주스) : ");
6     scanf("%d", &select);
7     switch (select) {
8     case 1:
9         printf("사과주스 3000원\n");
10        break;
11    case 2:
12        printf("딸기주스 3500원\n");
13        break;
14    case 3:
15        printf("키위주스 4000원\n");
16        break;
```

✓ 메뉴를 선택하는 형태의 프로그램을 구성하는데 switch문을 주로 이용

### 4.3 조건문 switch

```

17     case 4:
18         printf("오렌지주스 2500원\n");
19         break;
20     default:
21         printf("없는 메뉴입니다.\n");
22     }
23     return 0;
24 }

```

입력 예	2
출력 예	메뉴 선택 ( 1:사과주스 2:딸기주스 3:키위주스 4:오렌지주스 ) : 2 딸기주스 3500원.

### 4.3 조건문 switch

소스 4-11

```

1  #include <stdio.h>
2  int main(void)
3  {
4      char key;
5      scanf("%c", &key);
6      switch (key) {
7          case 'a':
8              printf("Attack!\n");
9              break;
10         case 'q':
11             printf("Spear!\n");
12             break;

```

### 4.3 조건문 switch

```

13     case 'w':
14         printf("Shield!\n");
15         break;
16     case 'e':
17         printf("Assault!\n");
18         break;
19     case 'r':
20         printf("Starfall!\n");
21         break;
22     }
23     return 0;
24 }

```

입력 예	q
출력 예	Spear!

### 4.3 조건문 switch

#### 소스 4-12

```

1  #include <stdio.h>
2  int main(void)
3  {
4      int op1, op2;
5      char op;
6      scanf("%d %c %d", &op1, &op, &op2);
7      switch (op) {
8          case '+':
9              printf("%d %c %d = %d\n", op1, op, op2, op1+op2);
10             break;
11          case '-':
12              printf("%d %c %d = %d\n", op1, op, op2, op1-op2);
13             break;
14          case '*':
15              printf("%d %c %d = %d\n", op1, op, op2, op1*op2);
16             break;

```



### 4.3 조건문 switch

```

17     case '/':
18         if(op2!=0) printf("%d %c %d = %d\n", op1, op, op2, op1/op2);
19         else printf("Can not divide by 0!\n");
20         break;
21     case '%':
22         printf("%d %c %d = %d\n", op1, op, op2, op1%op2);
23         break;
24     default:
25         printf("Wrong operator!\n");
26     }
27     return 0;
28 }

```

입력 예 1	7 % 2
출력 예 1	7 % 2 = 1
입력 예 2	5 ^ 2
출력 예 2	Wrong operator!
입력 예 3	10 / 0
출력 예 3	Can not divide by 0!

### 4.3 조건문 switch

- switch 문에서 break를 의도적으로 사용하지 않는 경우
  - ✓ 여러 개의 case를 하나로 묶어서 프로그램을 실행하고자 하는 경우
  - ✓ break문을 생략 시, 해당하는 case문 이후의 모든 문장들을 모두 처리
  - ✓ break문을 만나지 않는 경우, switch 문 내의 모든 문장 실행

### 4.3 조건문 switch

소스 4-13

```

1  #include <stdio.h>
2  int main(void)
3  {
4      int score;
5      scanf("%d", &score);
6      switch (score/10) {
7          case 10:
8          case 9:
9              printf("%d점은 A입니다.\n", score);
10             break;
11         case 8:
12             printf("%d점은 B입니다.\n", score);
13             break;

```

break문 생략

### 4.3 조건문 switch

```

14         case 7:
15             printf("%d점은 C입니다.\n", score);
16             break;
17         case 6:
18             printf("%d점은 D입니다.\n", score);
19             break;
20         default:
21             printf("%d점은 F입니다.\n", score);
22             break;
23     }
24     return 0;
25 }

```

입력 예	65
출력 예	65점은 D입니다.

### 4.3 조건문 switch

소스 4-14

```

1  #include <stdio.h>
2  int main(void)
3  {
4      int month;
5      scanf("%d", &month);
6      switch (score/10) {
7          case 3: case 4: case 5:
8              printf("Spring\n", score);
9              break;
10         case 6: case 7: case 8:
11             printf("Summer\n", score);
12             break;

```

### 4.3 조건문 switch

```

13         case 9: case 10: case 11:
14             printf("Fall\n", score);
15             break;
16         case 12: case 1: case 2:
17             printf("Winter\n", score);
18             break;
19         default:
20             printf("Wrong month\n", score);
21             break;
22     }
23     return 0;
24 }

```

입력 예	10
출력 예	Fall

#### 4.4 Q&A

Q 17. 어떤 사람들은 `if(x==0)`을 쓰지 않고 `if(0==x)`와 같은 코드를 쓰는데, 그 이유는 무엇인가?

A. 다음과 같은 실수를 하는 것을 막기 위함이다.

`if(x = 0)`

상수를 `==` 연산자의 왼쪽에 쓰는 습관을 들이면, 다음과 같은 실수를 했을 때, 컴파일러에서 에러가 발생한다. :

`if(0 = x)`

어떤 사람들은 이런 방식을 쓰는 것을 외우는 것보다 두 개의 `==`를 쓰는 것을 외우는 것이 쉽기 때문에, 이러한 방식을 좋아하지 않는다. (operand가 상수일 경우에만 도움이 됨)

#### 4.5 실습 및 과제

- 실습 및 과제 진행
  - DCU Code : <http://code.cu.ac.kr>
- 4 주차 실습
  - 코딩 4-1, 코딩 4-2, 코딩 4-3, 코딩 4-4,  
코딩 4-5, 코딩 4-6, 코딩 4-7, 코딩 4-8 (p166-180)
- 4 주차 과제
  - 4장 프로그래밍 연습 1-4번 (p102)



#### 4.6 참고 문헌

---

- 컴퓨터 프로그래밍 : [https://ko.wikipedia.org/wiki/컴퓨터\\_프로그래밍](https://ko.wikipedia.org/wiki/컴퓨터_프로그래밍)
- 어셈블리어 : <https://ko.wikipedia.org/wiki/어셈블리어>
- C Programming FAQ : <http://cinsk.github.io/ko/cfaqs/index.html>

END

