

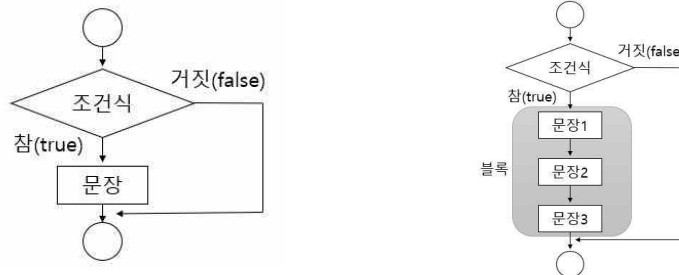
## ■ 1교시 학습활동: 조건문, 반복문, 제어문 학습정리

**조건문:** 조건식의 결과가 참, 거짓인지에 따라 분기하여 실행하는 문장이 다른 것으로, (0을 제외한) 정수값은 참으로 판별하고, 0이면 거짓으로 판별한다.

조건문의 종류로는 if 문, if~else 문, 중첩 if~else 문, 다중 if~else 문, switch 문 등이 있다.

### 1. if 문

하나의 단일 if 문으로 구성되고, if 문에 사용된 제어문의 조사 결과에 따라 문장의 실행 여부를 결정한다. 괄호 안의 조건식이 참이면 옆의 문장을 실행하고, 거짓이면 문장을 실행하지 않고 다음 문장으로 넘어간다.



### <단순 if 문의 복잡한 조건 출력 실습>

**Annotation 1:** x = 9이기 때문에 조건문은 참이 되어 다음 출력문을 출력한다.

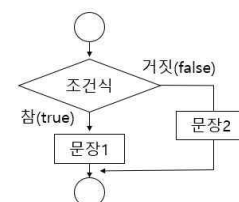
**Annotation 2:** ++value % 2 != 0 value의 값이 8인 상태에서 ++을 해줬기 때문에 value의 값은 9가 된다. 9/2 연산 결과값은 4.5 이기에 0이 아니라는 조건을 만족(참)하게 된다.

**Annotation 3:** ++count < limit Count의 값이 10인 상태에서 ++을 해줬기 때문에 count의 값은 11이 된다. Limit와 비교하면 11 < 11 이므로 연산이 맞지 않는다. 따라서 조건을 만족하지 못하게 (거짓) 된다..

**Annotation 4:** OR연산자가 사용되어 하나의 조건만 충족되어도 if문이 실행된다.

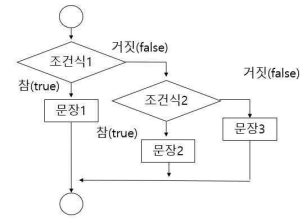
### 2. if ~ else 문

제어문의 판별 결과에 따라 참이면 “문장 1”을, 거짓이면 “문장 2”로 분기되는 제어문이다. (if ~ else 문에서 else 문 사이의 문장이 한 문장일 경우, { } 표시를 따로 하지 않아도 된다. 반면에 여러 문장일 경우, { }을 표시해주어야 한다. 생략할 시 에러가 발생.)



## 2-1. 중첩 if ~ else 문 (중첩 if)

중첩 if 문장은 비규칙적인 요소를 비교하는 데 있어서 복잡한 환경을 제공하지만, 규칙적인 요소를 제공한다. 조건에 맞는 문장을 계속 체크하여 내려가고, 만약 조건이 맞을 경우, 해당 문장을 수행하고 if ~ else ~ if 문장을 빠져나간다. (첫 번째 if 문에서 “조건식 1”이 참이면 “문장 1”을, 거짓이면 else 문의 “조건식 2”를 평가해서 그 식이 참이면 “문장 2”를, 거짓이면 “문장 3”을 실행한다.)



### <입력한 점수의 학점 구하기 실습>

사용자가 입력한 점수가 95이기 때문에  $90 \leq 95$ 라는 조건이 만족되어, 95점 이므로 A학점입니다.라는 문장이 출력된다.

사용자가 입력한 점수에 따라 조건문이 실행된다. 조건을 만족하면 바로 다음의 문장을 출력하고, 조건을 만족하지 못하면 다음 else if 문으로 차례대로 넘어가 조건을 체크한다.

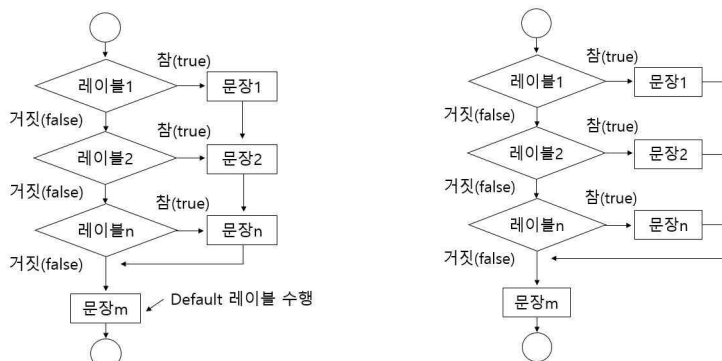
```

1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int score;
6     cout << "점수를 입력하십시오 : ";
7     cin >> score;
8
9     if (90 <= score && score <= 100)
10        cout << score << "점이므로 A학점입니다." << endl;
11     else if (80 <= score && score <= 89)
12        cout << score << "점이므로 B학점입니다." << endl;
13     else if (70 <= score && score <= 69)
14        cout << score << "점이므로 C학점입니다." << endl;
15     else if (60 <= score && score <= 59)
16        cout << score << "점이므로 D학점입니다." << endl;
17     else
18        cout << score << "점이므로 F학점입니다." << endl;
19 }
    
```

출력: 점수를 입력하십시오 : 95  
95점이므로 A학점입니다.

## 3. switch 문

switch 문은 조건문의 변수나 수식을 평가한 뒤에 그 결과값에 따라 case 다음 식을 비교하여 값이 같은 곳의 명령문을 수행한다. 수행한 후에 break 문을 만나는 곳에서 switch 문은 종료하게 되는데, 만약 일치하는 것이 없다면 default 문을 수행하게 된다. (case 문 끝에 break 문이 없으면 계속해서 다음 case 문들을 수행하기 때문에 break 문을 반드시 코드에 작성해야 한다.)



### 3-1. 중첩 switch 문

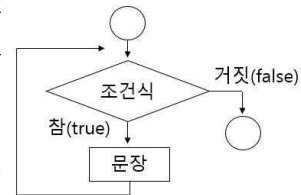
switch 문장을 사용할 때, switch 블록 안에서 별도의 switch 문장을 사용한다.

**반복문:** 특정한 문장을 반복하여 실행시켜야 할 경우에 사용하는 것으로, 어떤 문장을 반복하려면 처음 시작하는 초기값과, 반복을 마치는 조건식, 반복 값을 증가시키거나 감소시키는 감소식이 필요하다. 조건이 참인 동안 블록 안의 실행문을 처리한다.

반복문의 종류는 for 문, while 문, do-while 문이 있다.

#### 1. while 문

괄호 안에 선언된 조건문을 판별하여 참인 경우, 블록 안의 실행문을 처리하고 다시 조건문을 반복하여 판별한다. 조건문의 결과가 거짓일 경우, while 블록을 벗어난다. 조건문의 결과가 처음부터 거짓일 경우 반복 문장을 단 한 번도 실행하지 않는다. (0 이외의 값이 참이고, 0이면 거짓이다. 무한 반복코드는 while(1)로, 빠져나올 때는 break, return 문이 필요하다. 또한, while 문에서 switch 문과 같이 내포된 while 문을 사용할 수 있다.)



#### <중첩 while 블록의 연산 결과 실습>

index는 1, 1 <= 9 라는 조건을 만족하기 때문에 다음 블록 내의 문장을 실행한다.

```

1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int index = 1;
6     cout << "WwWwWwWw 구 간 " << endl;
7     cout << "WwWwWwWw " << endl;
8     while (index <= 9) {
9         int in = 2;
10        cout << "in " << endl;
11
12        while (in <= 9) {
13            int mult = in * index;
14            cout << in << "X" << index << " = "
15            << ((mult >= 10) ? "" : " ") << mult << " ";
16
17            in = in + 1;
18        }
19
20        index = index + 1;
21        cout << " " << endl;
22    }
23 }
```

in은 2, 2 <= 9 라는 조건을 만족하기 때문에 다음 블록 내의 문장을 실행한다.  
`mult = 2 * 1` 연산, `mult >= 10` 일 때는 " ", 그렇지 않을 경우 " ".  
`in`의 값을 1씩 증가시켜 9단까지 곱셈이 출력되도록 반복한다.

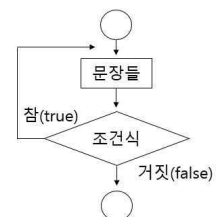
in이 9가 될 때까지 반복한 후, while문을 끝내고, `index = index + 1` 연산을 하여, 곱해지는 수가 9가 될 때까지 (거짓이 될 때까지) 반복문을 계속 실행한다.

```

2X1= 2 3X1= 3 4X1= 4 5X1= 5 6X1= 6 7X1= 7 8X1= 8 9X1= 9
2X2= 4 3X2= 6 4X2= 8 5X2=10 6X2=12 7X2=14 8X2=16 9X2=18
2X3= 6 3X3= 9 4X3=12 5X3=15 6X3=18 7X3=21 8X3=24 9X3=27
2X4= 8 3X4=12 4X4=16 5X4=20 6X4=24 7X4=28 8X4=32 9X4=36
2X5=10 3X5=15 4X5=20 5X5=25 6X5=30 7X5=35 8X5=40 9X5=45
2X6=12 3X6=18 4X6=24 5X6=30 6X6=36 7X6=42 8X6=48 9X6=54
2X7=14 3X7=21 4X7=28 5X7=35 6X7=42 7X7=49 8X7=56 9X7=63
2X8=16 3X8=24 4X8=32 5X8=40 6X8=48 7X8=56 8X8=64 9X8=72
2X9=18 3X9=27 4X9=36 5X9=45 6X9=54 7X9=63 8X9=72 9X9=81
  
```

#### 2. do ~ while 문

while 문과 유사한 기능을 가졌지만, do ~ while 문은 마지막에 조건식을 판별하여 적어도 do 문 아래의 블록 문장을 한번 수행한다. 조건식을 판별해서 참일 경우 do 문부터 다시 시작하고, 거짓일 경우 do ~ while 문을 빠져나간다.



## <do ~ while 문과 switch 문의 혼합 사용 실습>

do-while문은 적어도 한 번은 실행하는 반복문이다. 때문에 다음의 블록 내의 문장들을 한번 실행한다.

Switch문이 사용되어, 번호 선택 문장이 출력된 후, 번호를 입력하면 다음의 번호를 가진 케이스와 케이스 내의 블록이 실행된다.

코드의 줄이 너무 길어서, 중요한 do-while 문 부분만 캡처 했습니다!

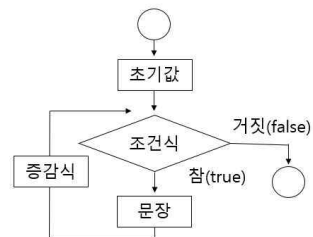
```

1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     char choice;
6     do {
7         cout << "제어/반복문" << endl;
8         cout << "1. if" << endl;
9         cout << "2. switch" << endl;
10        cout << "3. while" << endl;
11        cout << "4. do-while" << endl;
12        cout << "5. for" << endl;
13        cout << "번호 선택" << endl;
14        cin >> choice;
15    } while (choice < '1' || choice > '5');
16
17    cout << "\n" << endl;
18
19    switch (choice) {
20        case '1':
21            cout << "if 문\n" << endl;

```

## 3. for 문

반복 횟수가 명확하게 정해진 경우에 사용하는 반복문이다. 수식을 정의하는 초기값, 반복 횟수를 정의하는 조건식, 그리고 초기값에서 주어진 변수의 값이 증가되거나 감소하는 증감식이 사용된다. 초기값으로 시작되어 조건식이 참이 되는 동안만 반복하고, 거짓이 되면 블록을 빠져나간다. 한번 실행될 때마다 증감식이 연산된다. (무한 반복 코드는 for( ; ; )이고, 빠져나올 때 제어문이 필요하다. 또한, 때에 따라서 ,를 이용하여 초기값과 증감식을 다중으로 정의할 수 있다.)



### 3-1. 중첩된 for 문

for 문의 블록 안에 중첩된 for 문을 선언할 수 있다. 내부의 for 문과 외부의 for 문은 같싸 있는 형태로 선언되어야 한다.

**제어문(분기문):** 프로그램이 특정한 목적에 따라 순차적인 흐름을 벗어나 특정 부분으로의 분기나 탈출 등이 요구될 때 사용하는 것이다.

제어문의 종류는 break, return, continue 문 등이 있다.

### 1. break 문

break 문은 switch 문장에서 문장의 흐름을 막아주는 데 사용되며, 반복문의 블록을 빠져나가는 데 자주 활용된다. 또한, 반복 블록 내에 선언되어 블록을 끝내주거나, 프로그램의 제어권을 다음 문장으로 넘겨 다시 시작해주는 역할을 한다. (break 문은 if 문에 의한 블록에는 적용되지 않으며, 블록 문이 하나 이상 중첩된 반복문에서 사용된 break 문은 가장 안쪽의

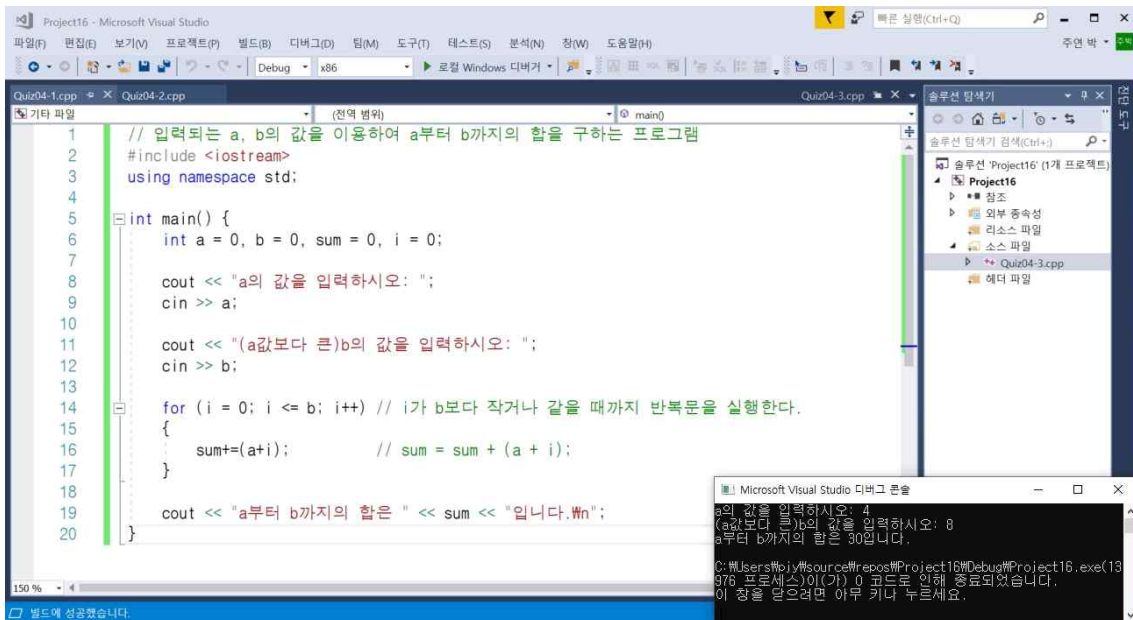


블록을 깨고 탈출한다.)

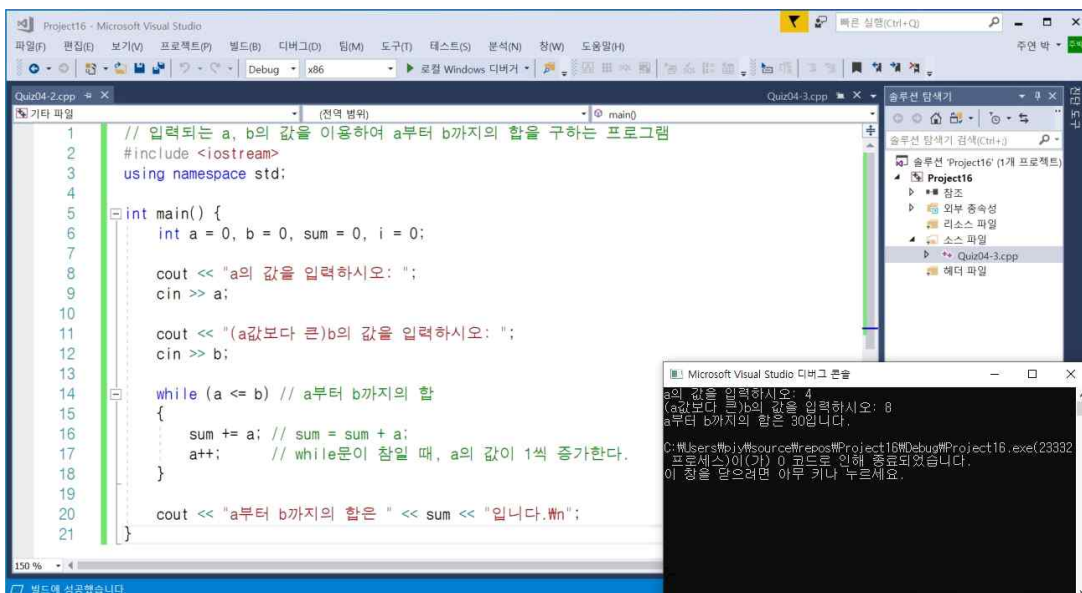
## 2. continue 문

loop 내의 문장을 더 이상 실행하지 않고, 다음 과정으로 이동하는 제어문이다. (for 문 - 조건의 증감식을 다시 수행, while 문&do~while 문 - 조건식을 다시 체크한다. 따라서 continue 문을 만나는 지점에서 블록을 한번 수행한 것으로 간주한다.

- 2교시 학습활동: for 문을 이용하여 두 정수 a, b를 입력받아 a에서 b까지의 합을 출력하라. (단, 작은 수, 큰 수 순서로 입력받는다. ex-4,8)



- 3교시 학습활동: 1번의 for 문을 while 문으로 바꾸어 출력하라.



■ 4교시 학습활동: 1번의 for 문을 do-while 문으로 바꾸어 출력하라.

