Chapter5\_조건문

학습목표

1. if문을 이해하고 사용할 수 있다.
2. If-else문을 이해하고 사용할 수 있다.
3. 중첩 if문을 이해하고 사용할 수 있다.
4. switch문을 이해하고 사용할 수 있다.
5. break, continue, goto문을 이해하고 사용할 수 있다.
6. 3가지의 기본 제어구조

순차구조(sequence)-명령어들이 순차적으로 실행되는 구조이다.

선택구조(selection)-둘 중의 하나의 명령어를 선택하여 실행되는 구조이다.

반복구조(iteration)-동일한 명령어가 반복되면서 실행되는 구조이다.

1. If-else문

if ( 조건식 ) {

문장1;

}

else {

문장2;

}

설명: 만약 조건식이 참이면 문장1이 실행되고, 그렇지 않으면 문장2가 실행된다.

조건에 따라 실행되는 문장이 하나이면 중괄호는 생략이 가능하다. else이하는 생략될 수 있다.

1. 조건 연산자

If-else문과 유사한 기능을 하는 조건 연산자. C에서 유일하게 3개의 피연산자를 가지는 삼항 연산자이다.

형식: 조건식 ? 값1(참일 경우) : 값2(거짓일 경우)

조건식의 조건이 참이면 값1을 반환하고, 거짓일 경우 값2를 반환한다.

1. 복잡한 조건식을 사용하는 if-else문

복잡한 수식은 논리 연산자를 사용하여야 한다.

논리 연산자는 여러 개의 조건을 조합하여 참인지 거짓인지를 따질 때 사용한다.

Ex) x && y AND, x || y OR, !x NOT

논리 연산자를 포함한 수식을 작성할 때 주의해야 할 점이 있다. 여러 개의 피연산자 중에서 처음 피연산자의 값이 거짓이면 다른 피연산자들은 계산되지 않는다.

1. 연속적인 if문

if(조건식1) {

//조건식1이 참인 경우 실행할 명령문

}

else if(조건식2) {

//조건식2이 참인 경우 실행할 명령문

}

else {

//모든 조건식이 거짓 일 때 실행될 명령문

}

1. Switch문

가능한 실행 경로가 여러 개인 경우에는 if문보다 switch문을 사용하는 것이 좋다.

형식:

Switch(변수)

{

Case1 c1:

처리1;

Break;

Case c2:

처리2;

Break;

Case c3:

처리3;

Break;

Default:

처리4;

Break;

}

위에 변수라 했지만 switch문의 변수 자리에는 어떤 수식도 들어갈 수 있다.

여기서 주의할 점은 break문이 없으면 선택된 case절 안의 문장들을 실행한 다음, 계속해서 다음 case절의 문장들을 실행하게 된다. 따라서 case문은 일반적으로는 break문으로 끝내야 한다.

기본개념

1. 중괄호로 묶은 여러 개의 문장 = 복합문, 블록