

C++ 실습 4

실습 내용

- 이번 주 실습은 이론시간에 배웠던 제어문들을 사용하는 실습입니다.
- 총 4개의 cpp파일이 있고 하나씩 실행하며 확인하면 됩니다.
- 지난주와 마찬가지로 Git Pull 명령을 이용해 새로운 파일들을 내려받습니다.

▼ practice4

🔗 pr1_if_stmt.cpp

🔗 pr2_switch.cpp

🔗 pr3_for_stmt.cpp

🔗 pr4_while_stmt.cpp

실습 시작전 설정 변경

- 이번 실습에는 강의에서 설명한 것처럼 Ranged for문이라는 C++11에서만 유효한 부분이 사용됩니다.
- 따라서 컴파일러에서 빌드를 할 때 C++11표준을 사용하겠다고 표시하는 부분을 추가해 주어야 합니다.
- tasks.json 파일을 열어 build설정을 하는 부분의 args에 다음 항목을 추가합니다.
 - “-std=c++11”
- 이미 컴파일러에서 C++11 표준을 지원하는 경우 이 과정이 없이도 잘 동작합니다.

첫번째 실습

- pr1_if_stmt.cpp 파일을 실행합니다.
- 간단하게 if-else문이 실행되는 것을 확인해보는 실습입니다.
- a, b, c의 값을 보고 각각의 조건이 참인지 거짓인지 판단합니다.
- 출력되는 내용을 토대로 어떤 문장들이 실행되었는지 확인할 수 있습니다.
- 강의시간에 배운대로 위와 아래 두 if문은 사실 같은 동작을 하는 것을 확인해 봅니다.

```
int a = 1, b = 5, c = 3;

//두 if-else문의 실행결과 비교.
if(a > b) {
    cout << "a is greater than b" << endl;
} else if(b >= c) {
    cout << "b is greater than or equal to c" << endl;
} else {
    cout << "c is greater than b" << endl;
}

if(a > b) {
    cout << "a is greater than b" << endl;
} else {
    if(b >= c) {
        cout << "b is greater than or equal to c" << endl;
    } else {
        cout << "c is greater than b" << endl;
    }
}
```

첫번째 실습

- 첫번째 실습의 후반부는 강의시간에 설명했던 다양한 분기문에 관한 정보를 확인하는 것입니다.
- ternary operator를 이용한 ternary conditional expression,
- short circuiting,
- 조건식에 대입연산자를 쓰는 경우에 대해서 확인해보세요.

```
int global = 10;
int update() {
    global = 1;
    return 1;
}

//ternary operator
a = b > c ? 10 : 0; //b = 5, c = 3
cout << "a = " << a << endl;

//short circuiting
a = -1;
if(a > 0 && update() > 0) { //update() changes global to 1.
    cout << "condition is true" << endl;
} else {
    cout << "condition is false" << endl;
}
cout << "global = " << global << endl;

//condition with assignment
if((a=b-c)>0) //b = 5, c = 3
    cout << "a = " << a << endl;
```

두번째 실습

- 두번째 실습은 switch문의 동작을 확인하는 것입니다.
- 상수로 5가지의 과일이 선언되었습니다.
- 그 중 3가지를 판별해 출력해주는 switch문입니다.
- break와 default의 역할에 주의하며 다양한 값을 입력해 봅니다.

```
const int APPLE = 1;
const int ORANGE = 2;
const int PEACH = 3;
const int MELON = 4;
const int COCONUT = 5;

int fruit;
cout << "Input one of the numbers in 1~3: ";
cin >> fruit;

switch(fruit){
    case APPLE:
        cout << "Selected Fruit is Apple!" << endl;
        break;
    case ORANGE:
        cout << "Selected Fruit is Orange!" << endl;
        break;
    case PEACH:
        cout << "Selected Fruit is Peach!" << endl;
        break;
    default:
        cout << "?? nothing found." << endl;
}
```

두번째 실습

- 두번째 실습의 후반부는 break문을 의도적으로 배치하거나 빼서 여러 case를 묶는 예시입니다.
- 껍질을 벗겨 먹는(peel) 경우와 과일 자체를 잘라서(cut) 먹어야 하는 경우를 구분하여 메시지를 출력합니다.

```
//Without break
cout << "Input one of the numbers in 1~5: ";
cin >> fruit;

switch(fruit){
    case APPLE:
    case ORANGE:
    case PEACH:
        cout << "Let's peel the fruit!" << endl;
        break;
    case MELON:
    case COCONUT:
        cout << "Let's cut the fruit!" << endl;
}
```

세번째 실습

- 세번째 실습은 for문과 관련된 실습입니다.
- 배열 a의 원소들을 모두 합하는 예제를 for문과 ranged for문으로 작성하였습니다.
- sum으로 표시되는 값을 확인합니다.
- 배열의 크기를 구하는 부분이나 복합대입 연산자, 배열의 접근 등 이전 시간에 배웠던 부분도 다시 기억해 두세요.

```
int a[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
int sum = 0;
//basic for statement
int length = sizeof(a)/sizeof(int);
for(int i=0; i<length; i++) {
    sum += a[i];
}
cout << "sum = " << sum << endl;

//Ranged for statement
sum = 0;
for(int i : a) {
    sum += i;
}
cout << "sum = " << sum << endl;
```


세번째 실습

- 후반부는 중첩된 for문을 사용하는 예제입니다.
- 첫부분은 2차원 배열 x의 원소를 특정부분만 출력하는 코드입니다.
- i와 j 두 변수를 동시에 사용하는 것도 가능하다는 점을 확인해 보세요.
- 두번째 부분은 두 개의 for문을 이용하여 ‘#’를 원하는 형태로 출력하는 코드입니다.
- 첫번째 for문의 변수 i를 두번째 for문의 조건에 사용하는 방식을 확인해 봅시다.

```
//5x5 matrix
int x[5][5] = {
    {1, 2, 3, 4, 5},
    {1, 2, 3, 4, 5},
    {1, 2, 3, 4, 5},
    {1, 2, 3, 4, 5},
    {1, 2, 3, 4, 5}
};

for(int i=0, j=0; i<5 && j<5; i++, j++) {
    for(int n=0; n<i; n++)
        cout << " ";
    cout << x[i][j] << endl;
}

for(int i=0; i<3; i++) {
    for(int j=0; j<=i; j++) {
        cout << "#";
    }
    cout << endl;
}
```

```
/*
#
##
###
*/
```

네번째 실습

- 네번째 실습은 반복해서 입력을 받는 while문에 관련된 실습입니다.
- 두번째 while문의 조건이 만족되지 않아 무시되는 것을 확인해 보세요.
- do-while의 경우 이런 문제가 생기지 않습니다.

```
//while loop
string input = " ";
while(!input.empty()) {
    cout << "Input your string: ";
    getline(cin, input);
    cout << "your input is " << input << endl;
}

//초기 조건 문제 - 이 지점에서 !input.empty() = false임.
while(!input.empty()) {
    cout << "Input something: ";
    getline(cin, input);
    cout << "your input is " << input << endl;
}

cout << "Using do-while loop" << endl;
do{
    cout << "Input your string again: ";
    getline(cin, input);
    cout << "your input is " << input << endl;
} while(!input.empty());
```

실습 제출물

- 이번주는 실습을 각자 진행해보고 따로 출석인정을 위한 화면 캡처는 제출하지 않습니다.
- 이번주의 실습관련 출석은 과제 제출로 대체됩니다.

실습 정리

- 총 4개의 파일로 실습을 진행하였습니다.
- if-else와 switch를 사용해 조건에 따라 다른 코드를 실행하는 방법을 확인했습니다.
- for문의 기본적인 사용법도 배웠습니다.
- while문, do-while문이 동작하는 방식도 확인하였습니다.