C++ Programming(Exercise) 3

- Arne Kutzner
- Hanyang University / Seoul Korea

Logical Operators(&&, ||, !)

- && is a boolean operator with value True/False.
  - (a && b) is true if both a &b are true
- || is a boolean operator with value True/False.
  - (a || b) is true when either a or b is true.
- ! is an unary boolean operator, which only has 1 operand.
  - (!a) means not a.

Example(Logical Operators)

```
1
2   #include <iostream>
3   using namespace std;
4
5   int main( ) {
6
7       bool a = 4;
8       a = 10 > 3;
9       cout << "10>3 is " << a << endl;
10      a = 13 <= 3;
11      cout << "13 <= 3 is" << a << endl;
12      a = (3 < 10) && (4 < 3);
13      cout << "(3<10) && (4<3) is " << a << endl;
14      a = (3 < 10) || (4 < 3);
15      cout << "(3<10) || (4<3) is " << a << endl;
16
17      a = !4;
18      cout << "!4 is " << a << endl;
19
20
21      return 0;
22
```

While structure

- While supports repetition. Same operation wil be repeated until the condition is false.
  - While(condition) { Do Something}

Example(While)

```cpp
#include <iostream>
using namespace std;


int main( ) {
    int num, i;
    int sum;

    cout << "Enter a positive number: ";
    cin >> num;

    i = 1;
    sum = 0;

    while (i <= num) {
        sum = sum + i;
        i = i++;
    }
    cout << sum;


    return 0;



}
```

If structure

- If structure is for a certain action should happen when a condition is true.
  - if(condition) {action}

Example(If structure)

```cpp
#include <iostream>
using namespace std;

int main() {


    char c;
    int total = 0;
    int spaces = 0;
    cin.get(c);//문자만 입력받는 함수

    while (c != '.') {
        if (c == ' ')
            ++spaces;
        ++total;
        cin.get(c);
    }

    cout << spaces << "spaces, " << total << "characters total in sentence" << endl;

    return 0;



}//end
```

If-else Structure

- If-else structure can specify an alternative action

```
if (condition)
    action if true
else
    action if false
```

Example

```
1
2   #include <iostream>
3   #include <stdlib.h>
4   using namespace std;
5
6   int main() {
7
8       int i, favn;
9       favn = rand();
10      cout << "Enter a favorite number from 1-100 to find ";
11      cout << "My favorite number: ";
12
13
14      do {
15          cin >> i;
16          if (i < favn)
17              cout << "Too low, guess again: ";
18          else if (i > favn)
19              cout << "Too high, guess again: ";
20          else
21              cout << favn << " is right!₩n";
22      } while (i != favn);
23
24      return 0;
25
```

Example(Switch)

```cpp
#include <iostream>
using namespace std;
enum {red, orange, yellow, green, blue}; //상수를 열거하는 방법

int main() {
    int code;
    cout << "Enter color code (0-4):";

    cin >> code;
    while (code >= red && code <= blue) {
        switch (code)
        {
        case red: cout << "Lips were red. \n"; break;
        case orange: cout << "Hair was orange. \n"; break;
        case yellow: cout << "Shoes were yellow. \n"; break;
        case green: cout << "Bag was green. \n"; break;
        case blue: cout << "Suit was blue. \n"; break;
        }
        cout << "Enter color code (0-4): ";
        cin >> code;
    }

    cout << "End. " << endl; //범위를 벗어날 시 출력


    return 0;


}
```

Do while structure

- Do while also provides repetition as while, but this time, condition is at the bottom loop.
  - It means, the body will be executed at least once.
  - do

```
    stuff;
while(condition);
```

Example(Do while)

```
#include <iostream>
using namespace std;


int main( ) {

    int i;

    cout << "Enter numbers in range 1~10 to find ";
    cout << "My favorite number " << endl;

    do
    {
        cin >> i;
    } while (i != 5);

    cout << "Yes, 5 is my favorite. " << endl;



    return 0;

}
```

For Structure

- For structure is used for loops that involve counting.
- You can use for loop as a while, and while as for.
  - for (initialization; condition ; update)
    do something