CONDITIONALS AND LOOPS

29 August 2023 10:02

CLASS 3rd — 28/08/23

https://www.linkedin.com/in/manojoffcialmj/

CONDITIONAL STATEMENTS

CONTROLS STATEMENTS:

Decision Making in programming is similar to decision making in real life. In programming too, we face some situations where we want a certain block of code to be executed when some condition is fulfilled.

Selection statements:

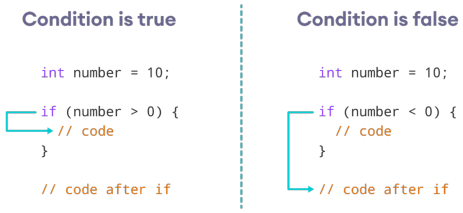
- if
- if-else
- if-else-if
- nested-if
- switch-case
- Ternary operator → Remaining

1. if STATEMENT

Syntax of if statement:

```
If (condition){
//statements
}
```

Working of if Statement:



Program of if statement:

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

int main(){
    int number = 10;
    // Check if number is greater than zero
    if(number > 0){
        cout << "Statement inside of if block" << endl;
    }
    cout << "Statement outside of if block" << endl;

    return 0;
}
```

Output:

```
Statement inside of if block
Statement inside of if block
```

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

int main(){
    int number = 10;
    // Check if number is greater than zero
    if(number < 0){
        cout << "Statement inside of if block" << endl;
    }
    cout << "Statement outside of if block" << endl;

    return 0;
}
```

Output:

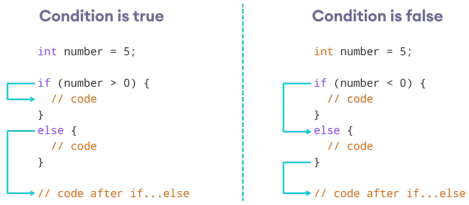
Statement inside of if block

2. If-else STATEMENT

Syntax of if-else statement:

```
If (condition){
//codes in if blocks}
else{
//codes in else blocks
}
```

Working of if-else Statement:



Program of if-else statement:

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

int main(){
    int number = 5;

    // Check if number is greater than zero
    if(number > 0){
        cout << "Statement inside of if block" << endl;
    }
    // Check if number is less than zero
    else{
        cout << "Statement inside of else block" << endl;
    }
    cout << "Statement outside of if-else block" << endl;

    return 0;
}
```

Output:

Statement inside of else block

Statements outside of if-else block

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

int main(){
    int number = -5;

    // Check if number is greater than zero
    if(number > 0){
        cout << "Statement inside of if block" << endl;
    }
    // Check if number is less than zero
    else{
        cout << "Statement inside of else block" << endl;
    }
    cout << "Statement outside of if-else block" << endl;

    return 0;
}
```

Output:

Statement inside of else block

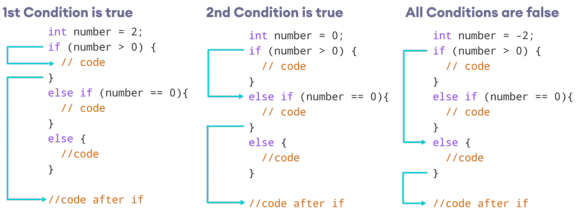
Statements outside of if-else block

3. If-else-if ladder STATEMENT

Syntax of if-else-if ladder statement:

```
If (condition1){
//codes
}
else if(condition2){
//codes
}
else if(condition2){
//codes
}
.
.
else {
//codes
}
```

Working of if-else-if ladder Statement:



Program of if-else-if ladder statement:

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

int main(){
    int number = 2;

    // Check if number is greater than zero
    if(number > 0){
        cout << "Statement inside of if block" << endl;
    }
    // Check if number is equal to zero
    else if (number == 0)
    {
        cout << "Statement inside of else-if block" << endl;
    }
    // Check if number is leass than zero
    else{
        cout << "Statement inside of else block" << endl;
    }
    cout << "Statement outside of if-else-if block" << endl;

    return 0;

}
```

Output:

Statement inside of if block
Statement outside of if-else-if block

4. Nested if-else STATEMENT

Syntax of nested if-else statement:

```
If (condition1)
{
        if(condition2)
        {
         //if codes
         }
        else
        {
         //else codes
         }
}
else
{
//codes
}
```

Program of nested if...else statement:

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

int main(){
    int number = 4;

    if(number == 4){
        if(number >= 10){
            cout << "Statement inside of nested if block" << endl;
        }
        else{
            cout << "Statement inside of nested else block" << endl;
        }
        cout << "Statement inside of if block" << endl;
    }
    else{
        cout << "Statement inside of else block" << endl;
    }
    cout << "Statement outside of if-else and nested if-else block" << endl;
    return 0;
}
```

Output:

Statement inside of nested else block
Statement inside of if block
Statement outside of if-else and nested if-else block

LOOP STATEMENTS

LOOPS:

Looping in programming languages is a feature that facilitates the execution of a set of instructions/functions repeatedly while some condition evaluates to true.

C++ provides three ways of executing the loops. While all the ways provide similar basic functionality, they differ in their syntax and condition checking the time.

Java's loops:

- for loop
- while loop
- do while loop
- for each loop

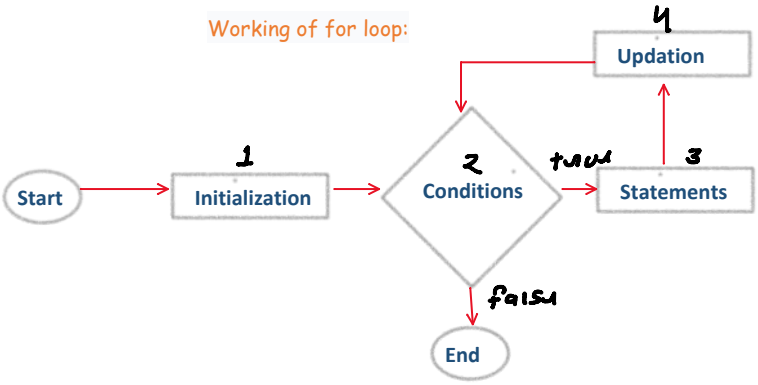
Remaining

1. for loop

Syntax of for loop:

```
for (initialization; conditions; updation)
{
// statements
}
```

Working of for loop:



Program of for loop:

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

int main(){
    // Print 5 time Babbar
    for(int i = 0; i<5; i=i+1){
        cout << "Babbar = " << i << endl;
    }

    return 0;
}
```

Output:

```
Babbar = 0
Babbar = 1
Babbar = 2
Babbar = 3
Babbar = 4
```

Nested for loop

Syntax of nested for loop:

```
for (initialization; conditions; updation)
{
    for (initialization; conditions; updation)
    {
        // statements
    }
}
```

outer loop

inner loop

Patterns problem are best understanding to nested for loop

I will provide this patterns problem in Homework with dry run.

Thanks MANOJ KUMAR