

1.3 conditions & loops

Made by :-

Akansha Jain



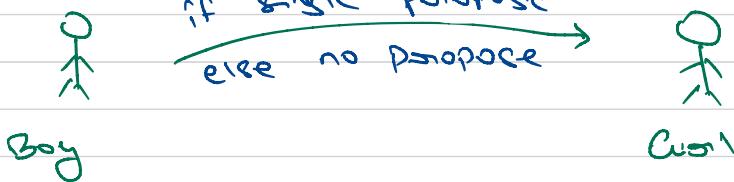
Decision Making

→ Conditional Statement



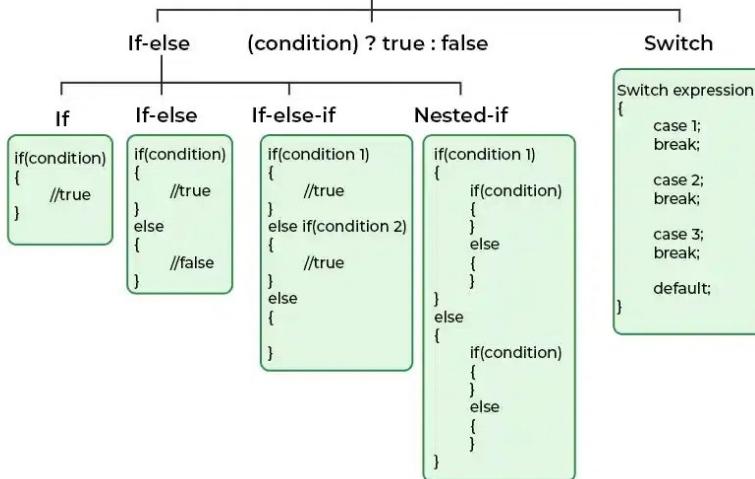
Used to make some decisions & based on these decisions we will execute the next block of code

also known as decision control structure



Types

Conditional Statements in C



Jump statement

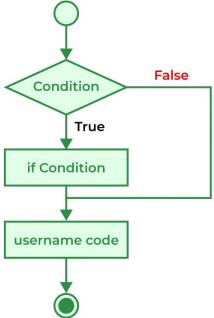
- break
- continue
- goto
- return

① If Statement

Condition after evaluation
will be true / false

Syntax :-

if (condition)
{



// Statement to execute if
// condition is true

}

If the condition is true,
block of code will execute
otherwise not

Ex

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

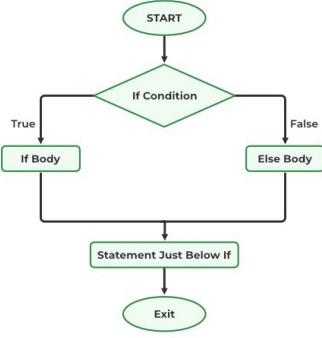
int main() {
    int x = 20;
    int y = 18;
    if (x > y) {
        cout << "x is greater than y";
    }
    return 0;
}
```

x is greater than y

Condition present in the if statement is
true, so, the block below the if
statement is not executed.

② If - else Statement

Syntax :-



If (Condition)

{

// Statement to execute if
// condition is true

}

else

{

// Statement to execute if
// condition is false

}

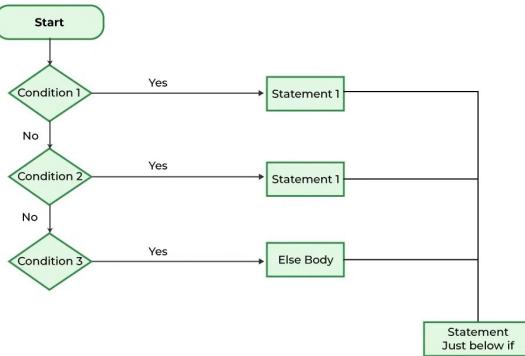
```
#include <iostream>
using namespace std;

int main() {
    int time = 20;
    if (time < 18) {
        cout << "Good day.";
    } else {
        cout << "Good evening.";
    }
    return 0;
}
```

Good evening.

③ If - else - if ladder

Syntax :-



if (Condition)

Statement ;

else if (Statement)

Statement ;

:

:

:

else

Statement ;

```
// Program to check whether a number is
positive, negative or 0 using if else if
ladder
#include <iostream>
using namespace std;

int main()
{
    int n = 0;

    // all Positive numbers will make
    // this condition true
    if (n > 0) {
        printf("Positive");
    }

    // all Negative numbers will make
    // this condition true
    else if (n < 0) {
        printf("Negative");
    }

    // if a number is neither Positive
    // nor Negative
    else {
        printf("Zero");
    }
    return 0;
}
```

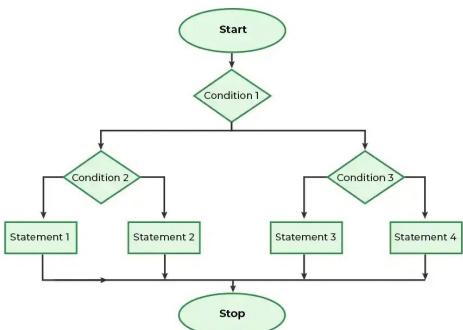
Zero

Q

Nested - if

Syntax :-

```
if (Condition 1)
    if (Condition 2)
        Statement ;
    else
        Statement ;
else
    Statement ;
```



```
// program to illustrate nested-if statement
#include <iostream>
using namespace std;

int main()
{
    int i = 10;

    if (i == 10) {

        // First if statement
        if (i < 15)
            cout << "i is smaller than 15\n";

        // Nested - if statement Will only be
        // executed if statement above is true
        else if (i < 12)
            cout << "i is smaller than 12 too\n";

        else
            cout << "i is greater than 15";
    }

    return 0;
}
```

i is smaller than 15

Special If Conditions

```
1) int n = 10;  
if (cout << n)  
{  
    cout << n;  
}
```

the condition in the
'if' condition is
true because it
ran successfully

Output \Rightarrow 10

```
2) int n; //n = 5  
if (cin >> n)  
{  
    cout << n;  
}
```

the condition in the
'if' condition is
true because it
ran successfully

Output \Rightarrow 5

Switch Case

Alternative to long if - else statement

Syntax

Switch (expression){

case value 1:

// Statement
break;

case value 2:

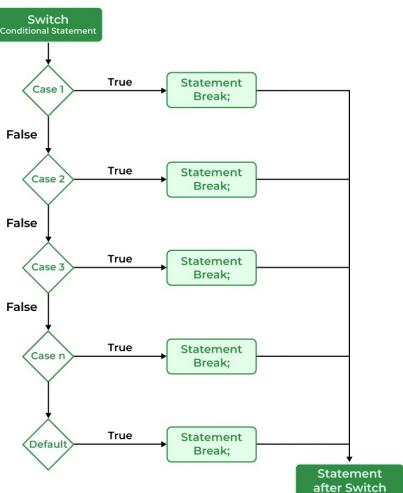
// Statement
break;

;
;
;
;

default:

// default statement
break;

}



Expression must be an integral value

that do not store
fractional values

↓
float, double

* Important Points

1) Switch → -ve , +ve allowed

Constant expression allowed

Switch (1+2+3)

Variable express allowed with fixed value

Switch (a+b+c)

2) Case → char & int allowed

→ -ve are allowed

→ Arithmetic operation

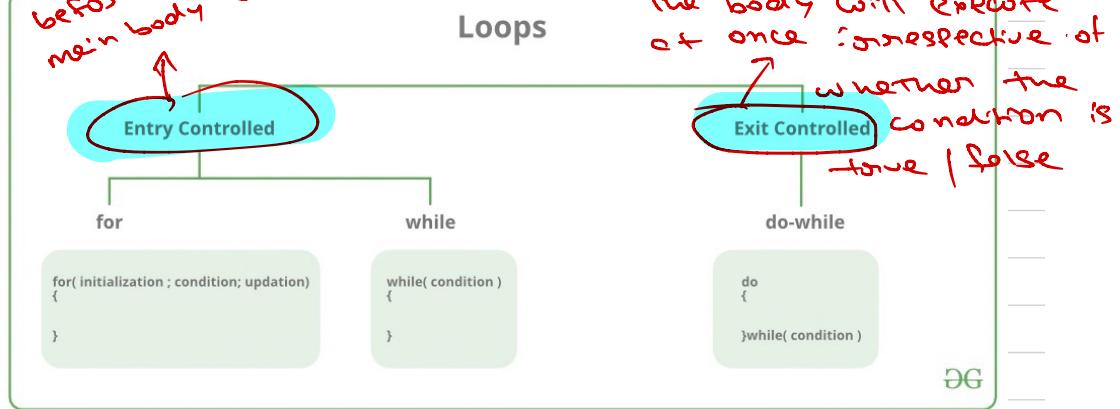
3) no duplicate case value allowed

4) If we using boolean in case then
it will not throw any error also
not print anything

Loops

→ used to repeat a block of code until the specified condition is met.

The test is checked before entering the main body of the loop



without loop

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

int main()
{
    cout << "Aditya Jain 1" << endl;
    cout << "Aditya Jain 2" << endl;
    cout << "Aditya Jain 3" << endl;
    cout << "Aditya Jain 4" << endl;
    cout << "Aditya Jain 5" << endl;

    return 0;
}
```

```
Aditya Jain 1
Aditya Jain 2
Aditya Jain 3
Aditya Jain 4
Aditya Jain 5
```

within loop

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

int main()
{
    for(int i = 1; i <= 5; i++)
    {
        cout << "Aditya Jain " << i << endl;
    }

    return 0;
}
```

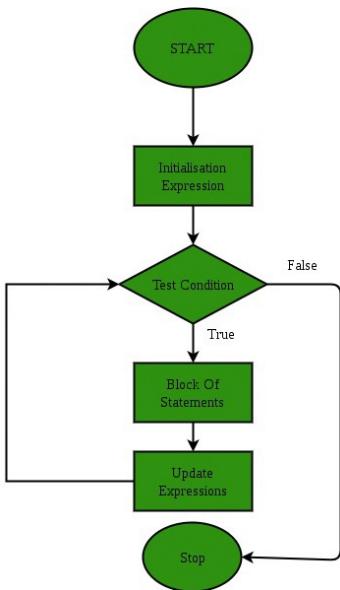
```
Aditya Jain 1
Aditya Jain 2
Aditya Jain 3
Aditya Jain 4
Aditya Jain 5
```

for loop

first initialize , then condition
check , then execute the body & at
last , the update is done

Syntax :-

for (initialize , condition , update)
{
 statement ;
}



```

#include <iostream>
using namespace std;

int main()
{
    for(int i = 1; i <= 5; i++)
    {
        cout << "Aditya Jain " << i << endl;
    }

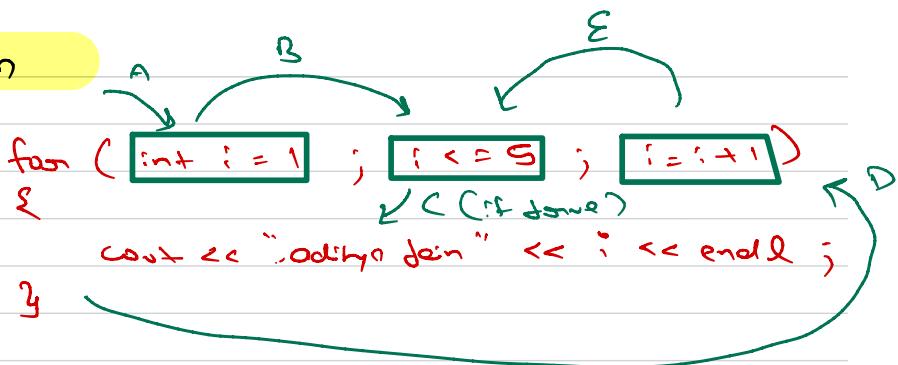
    return 0;
}
    
```

Red annotations have been added to the code to highlight specific parts: 'Initialization' points to the first part of the for loop's parentheses; 'Condition' points to the second part; and 'update' points to the third part.

```

Aditya Jain 1
Aditya Jain 2
Aditya Jain 3
Aditya Jain 4
Aditya Jain 5
    
```

Dry Run



Entry of loop?

$i = 1$



$i \leq 5$ (true)
"aditya join 1"
 $i = i + 1 \Rightarrow 1 + 1 \Rightarrow 2$



$i \leq 5$ (true)
"aditya join 2"
 $i = i + 1 \Rightarrow 2 + 1 \Rightarrow 3$



$i \leq 5$ (true)
"aditya join 5"
 $i = i + 1 \Rightarrow 5 + 1 \Rightarrow 6$

$i \leq 5$ (true)
"aditya join 4"
 $i = i + 1 \Rightarrow 4 + 1 \Rightarrow 5$

$i \leq 5$ (true)
"aditya join 3"
 $i = i + 1 \Rightarrow 3 + 1 \Rightarrow 4$



$i \leq 5$ (false)

Exit of loop

Special conditions

For loop without curly braces → The loop execute only one statement which is written just after it & the Statement cannot be declarative.

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

int main()
{
    int i;

    // for loop without curly braces
    for (i = 1; i <= 10; i++)
        cout << i << " ";
        cout << "\nThis statement executes after
for loop end!!!!" << endl; // Statement print only
once
    cout << "hi everyone" << endl;

    return 0;
}
```

only this statement in loop body

execute

```
1 2 3 4 5 6 7 8 9 10
This statement executes after for loop end!!!!
hi everyone
```

2) Infinite for loop / NULL Parameter loop

kind of for loop where the input parameters are not available or do not exist by which the loop iterates & runs endlessly.

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

int main()
{
    int num = 0;
    for (;;) // initialization, condition,
    updation are not entered
    {
        cout << "loop not end" << endl;
    }
    // Return statement to tell that everything
executed safely
    return 0;
}
```

```
loop not end
```

← never end

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

```
int main() {
    for (int i = 1; i <= 5; i = i - 1)
    {
        cout << i << endl;
    }
    return 0;
}
```

condition
never
false

```
1
0
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
-11
-12
-13
```

← never end but
Condition false never

Nested Loop

→ loop inside another
loop?

Syntax :-

// outer loop

for (initialization, condition, update)

{

// inner loop

for (initialization, condition, update)

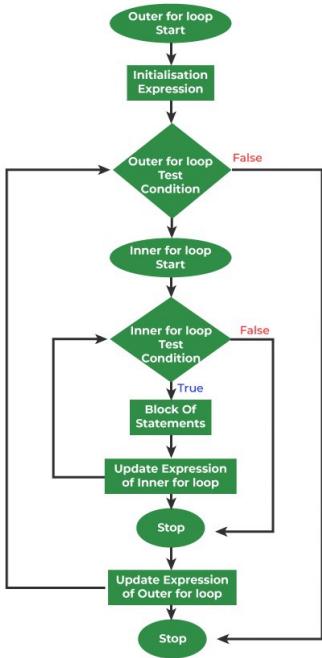
{

Statement of inner loop

}

Statement of outer loop

}



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

int main() {
    // Outer loop
    for (int i = 1; i <= 2; i=i+1) //Executes 2 times
    {
        cout << "Outer: " << i << endl;

        // Inner loop
        for (int j = 1; j <= 3; j=j+1)
        // Executes 6 times (2 * 3)
        {
            cout << " Inner: " << j << endl;
        }
    }
    return 0;
}
```

```
Outer: 1
Inner: 1
Inner: 2
Inner: 3
Outer: 2
Inner: 1
Inner: 2
Inner: 3
```