1.3 conditions & loops

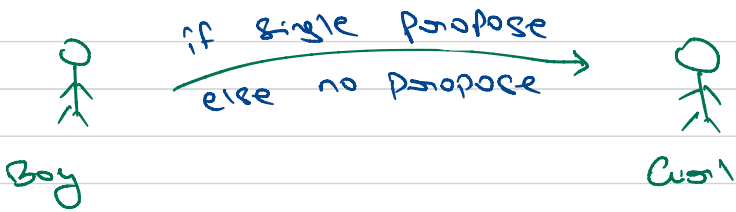
Made by :-

Aditya Jain

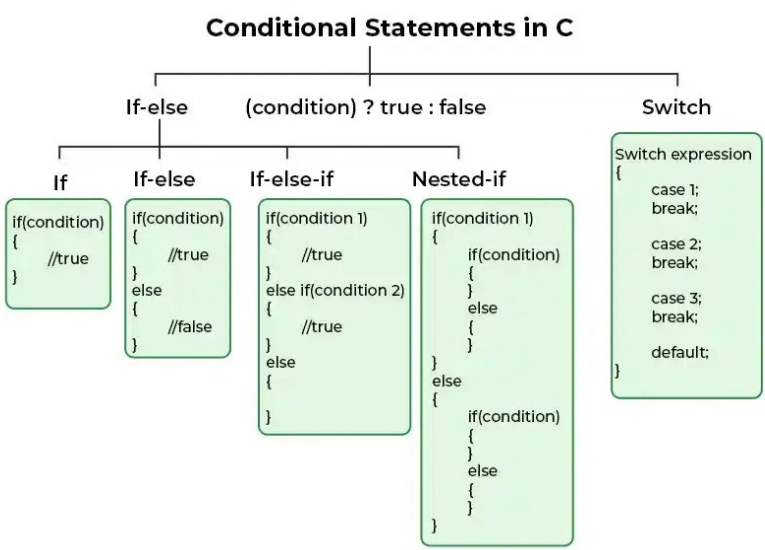
Decision Making → Conditional Statement

↓ also known as decision control structure

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



Types



Jump statement

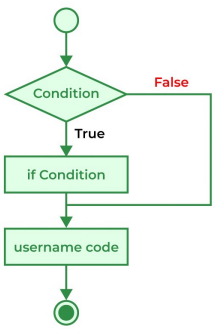
- break
- continue
- goto
- return

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

Exp

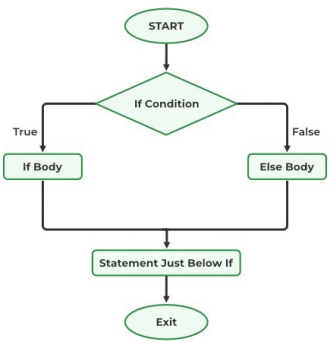
```
#include <iostream>
using namespace std;

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 true
}
```

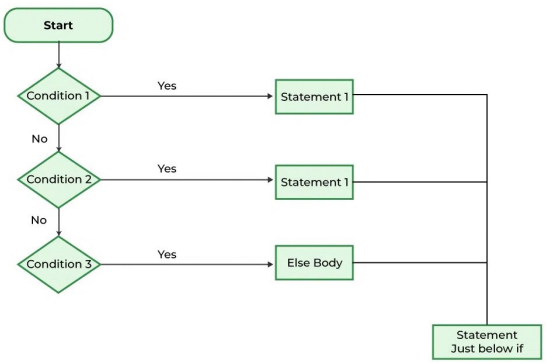
```cpp
#include <iostream>
using namespace std;

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 ;
```



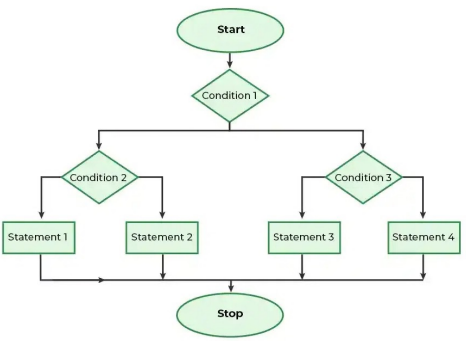

```cpp
// Program to check whether a number is positive, negative or 0 using if else if ladder
#include <iostream>
using namespace std;

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
```

4 Nested - if

Syntax:-

```
if (condition 1)
    if (condition 2)
        statement;
    else
        statement;
else
    statement;
```




```cpp
// 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 => 10

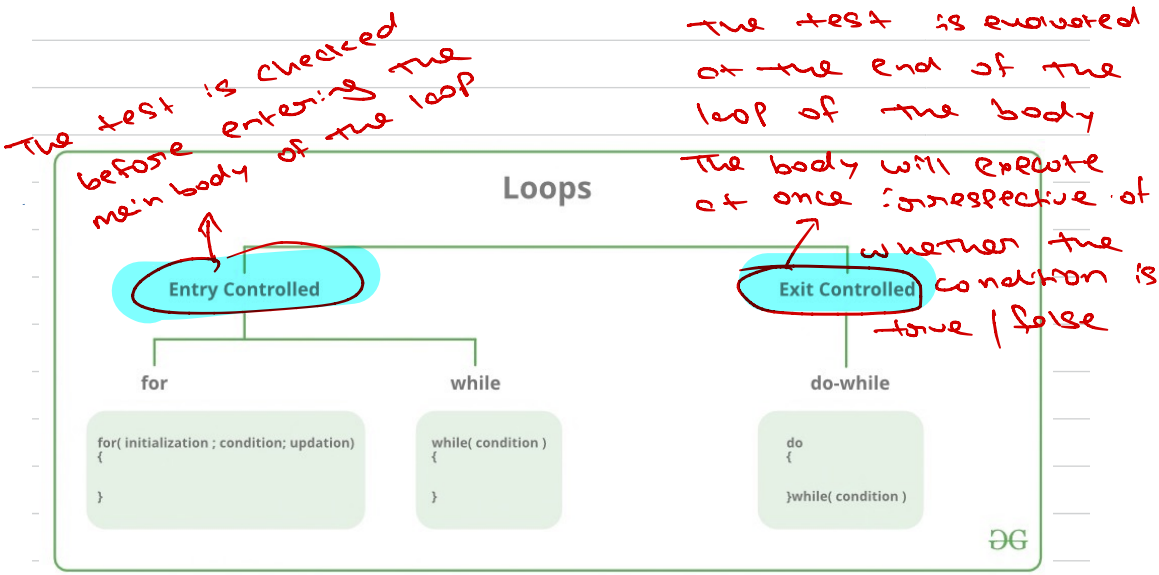
2)
```
int n; //n = 5
if ( cin >> n)
{
   cout << n;
}
```

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

output => 5

Loops

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



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
```

with loop

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

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

    return 0;
}
```

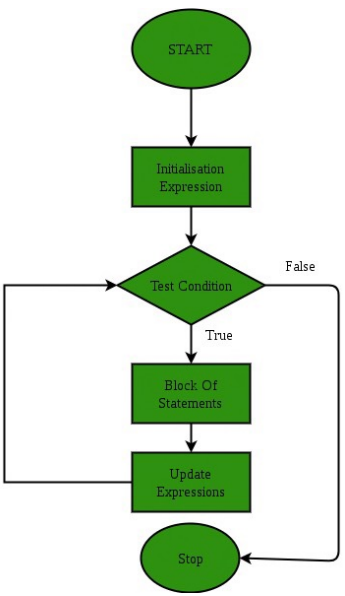
```
Aditya Jain 1
Aditya Jain 2
Aditya Jain 3
Aditya Jain 4
Aditya Jain 5
```

for loop

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

Syntax :-

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

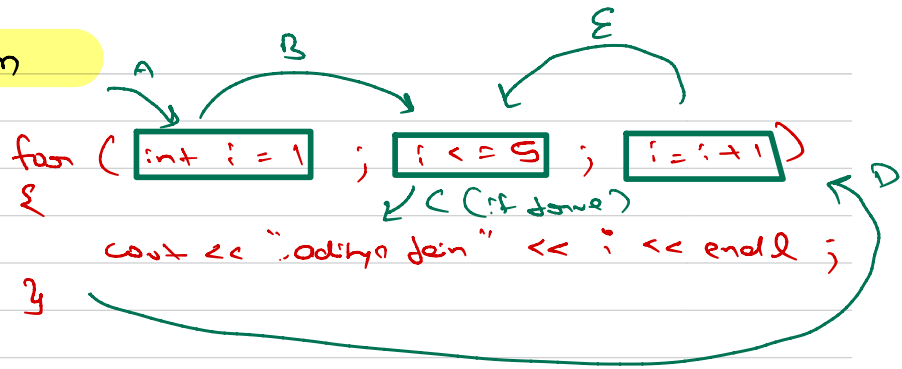



```cpp
#include <iostream>
using namespace std;

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

    return 0;
}
```

Initialization Condition Updation

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

Dry Run



Entry of loop?

i = 1

i <= 5 (true)
"aditya join 1"
i = i + 1 => 1 + 1 => 2

i <= 5 (true)
"aditya join 2"
i = i + 1 => 2 + 1 => 3

i <= 5 (true)
"aditya join 3"
i = i + 1 => 3 + 1 => 4

i <= 5 (true)
"aditya join 4"
i = i + 1 => 4 + 1 => 5

i <= 5 (true)
"aditya join 5"
i = i + 1 => 5 + 1 = 6

i <= 5 (false)

Exit of loop

Special conditions

⇒ for loop without curly bracket ⇒ The loop execute only one statement which is written just after it & the statement cannot be declarative.

```cpp
#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 execute in loop body

```
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 parameter are not available or do not exist by which the loop iterates & run endlessly.

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

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

```
loop not end
loop not end
loop not end
loop not end
loop not end
loop not end
loop not end
loop not end
loop not end
```

← never end

```cpp
#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
```

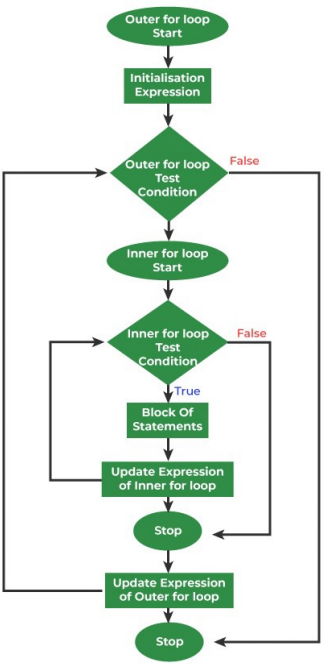
← never end bcoz condition never false

Nested Loop?

→ loop inside another loop?

Syntax :-

```
// outer loop
for (initialization , condition, updation)
{
    // inner loop
    for (initialization , condition, updation)
    {
        Statement of inner loop
    }
    Statement of outer loop
}
```




```cpp
#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
```