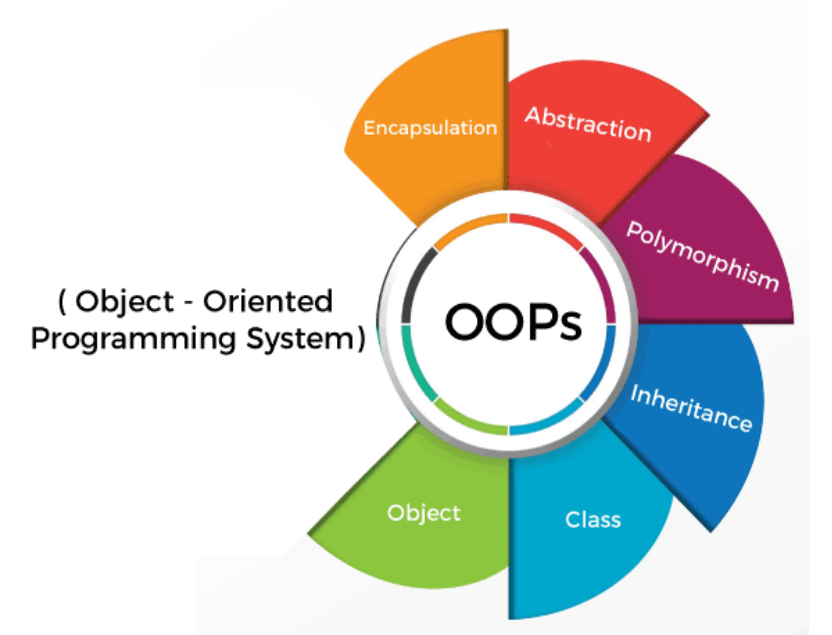
OOPS
OBJECT ORIENTED PROGRAMMING USING C++



1. INTRODUCTION TO CONTROL STATEMENTS
2. DIFFERENCE BETWEEN SIMPLE AND CONTROL STATEMENTS
3.EXPLAIN LOOPING STATEMENT

INTRODUCTION TO CONTROL STATEMENTS

- Control statements are used to control the flow of execution of a program.
- Any program can be more clear and understood if they use self-contained modules called as logic or control statements.
- They enable us to perform different actions based on certain conditions or loops.
- There are three types of control statements 1. Branching 2. Looping 3. Jumping

DIFFRENCE B/W SIMPLE AND CONTROL STATEMENTS

- Simple Statements
- A simple statement ends with a semicolon ; and cannot be further divided.
- For example:- #include<stdio.h>

```
main()
{
printf (“hello world”);
}
```

- Control statements
- Control statements are used to control the flow of execution of program.
- It combines instructions into logic unit.
- For example:-

```
void main(){
 int num=4;
 if(num>0){
  printf(“The number is
  positive\n”);
  }else{
  printf(“The number is
  negative\n”);
  }
```

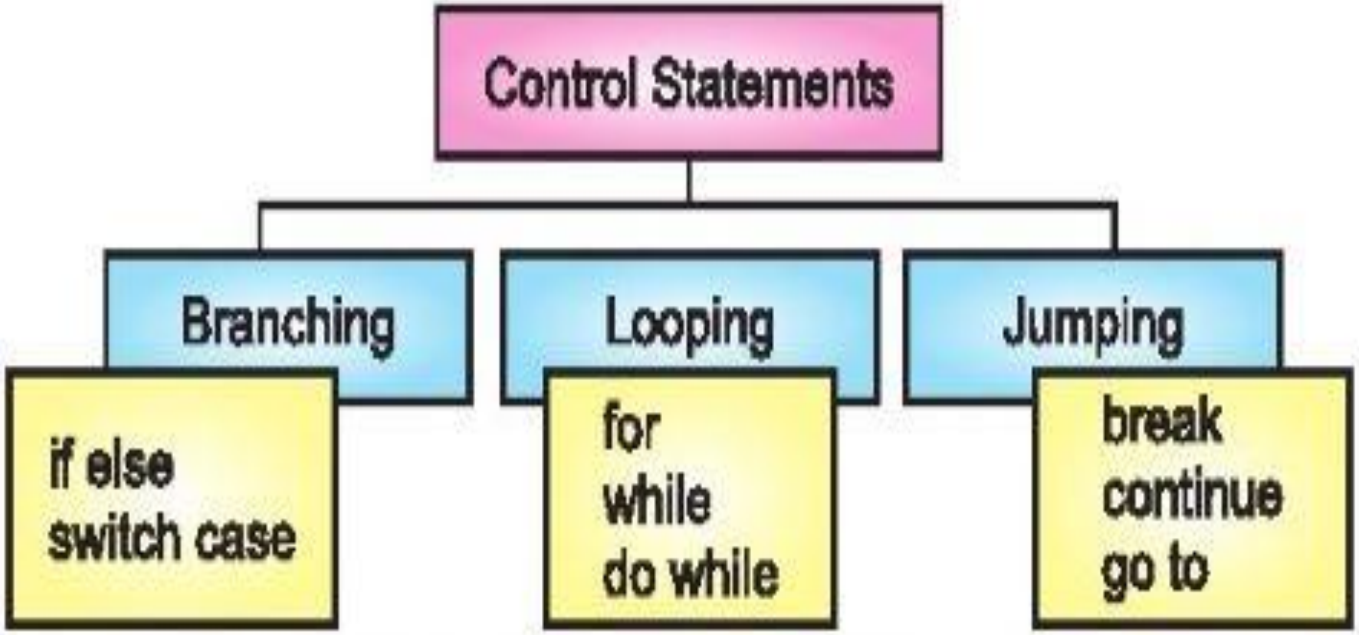


Fig. : 2.1 Different Types of Control Statements in C

LOOPING STATEMENT

Many problems require that a set of statements should be executed more than one time. Each time changing the values of one or more variables so that every execution if different from previous one. This kind of repetitive execution of a set of statements in a program is known as interactive Loop. There are three methods by which looping can be done by the program.

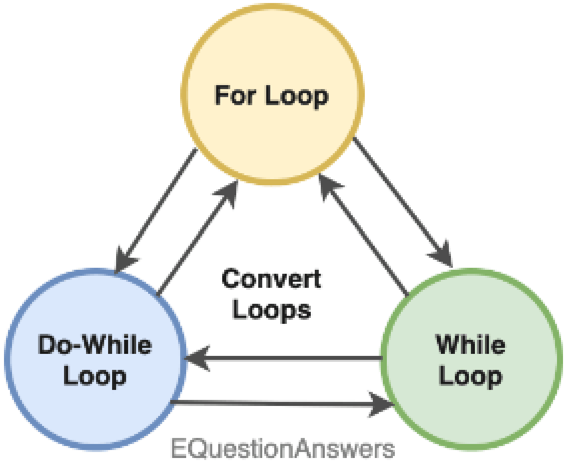
They are:- 1. Using a while statement

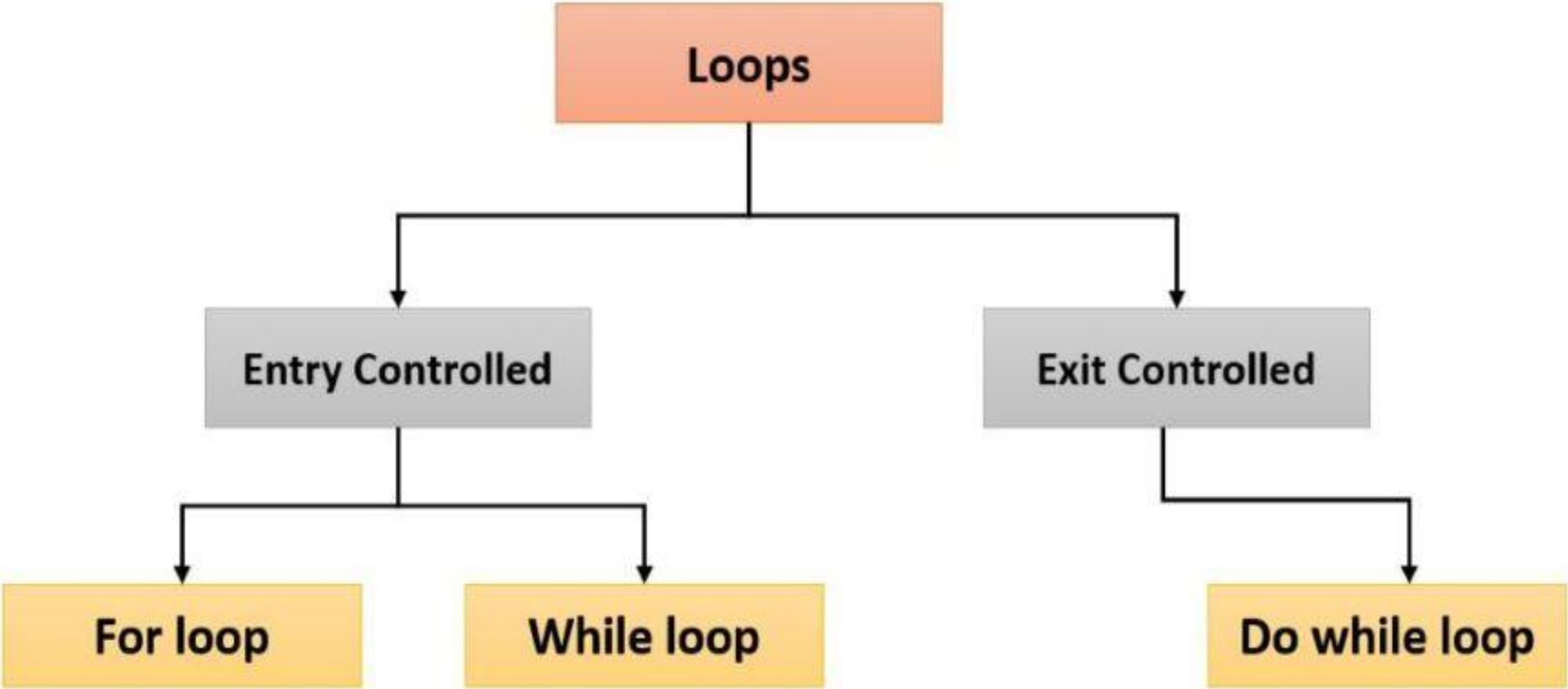
2. Using a for statement

3. Using a do-while statement

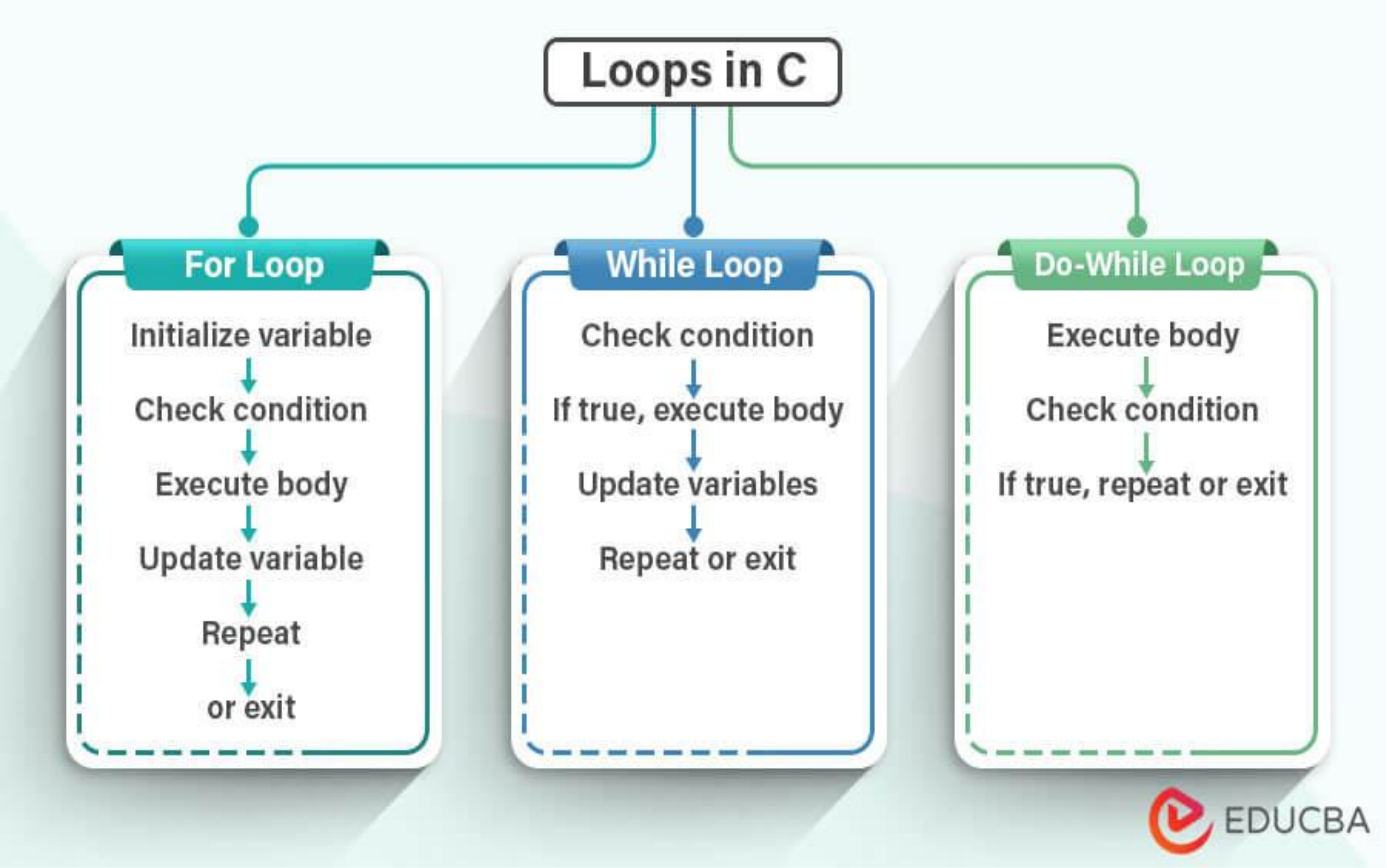
There are three types of loops in c++. They are:-

1. While loop
2. For loop
3. Do-while loop





Loops
Entry Controlled
Exit Controlled
For loop
While loop
Do while loop



Loops in C

For Loop

Initialize variable
↓
Check condition
↓
Execute body
↓
Update variable
↓
Repeat
↓
or exit

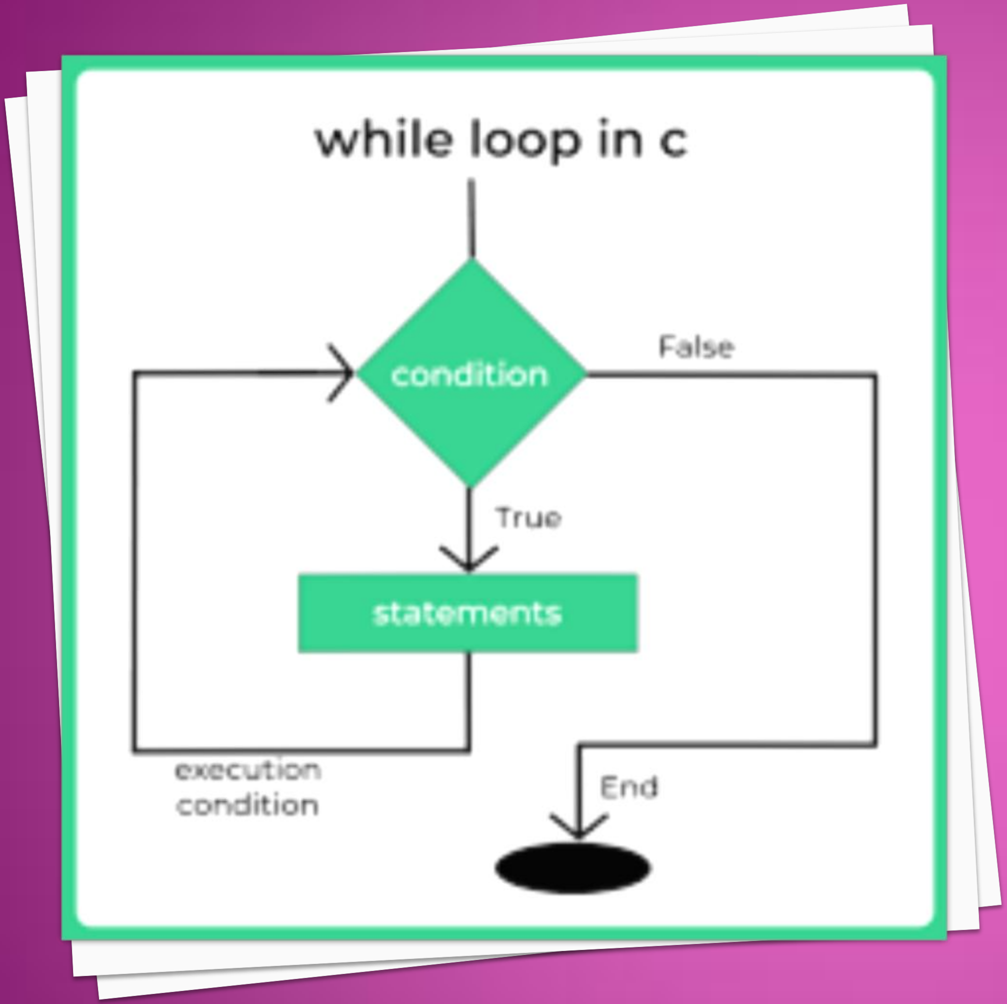
While Loop

Check condition
↓
If true, execute body
↓
Update variables
↓
Repeat or exit

Do-While Loop

Execute body
↓
Check condition
↓
If true, repeat or exit

EDUCBA



WHILE LOOP

It is the fundamental repetitive control structure used in c++.
While loop is suited for the problems where it is not known in advance that how many times statement or a set of statement i.e. statement block will be executed.
Syntax:- while(test expression)

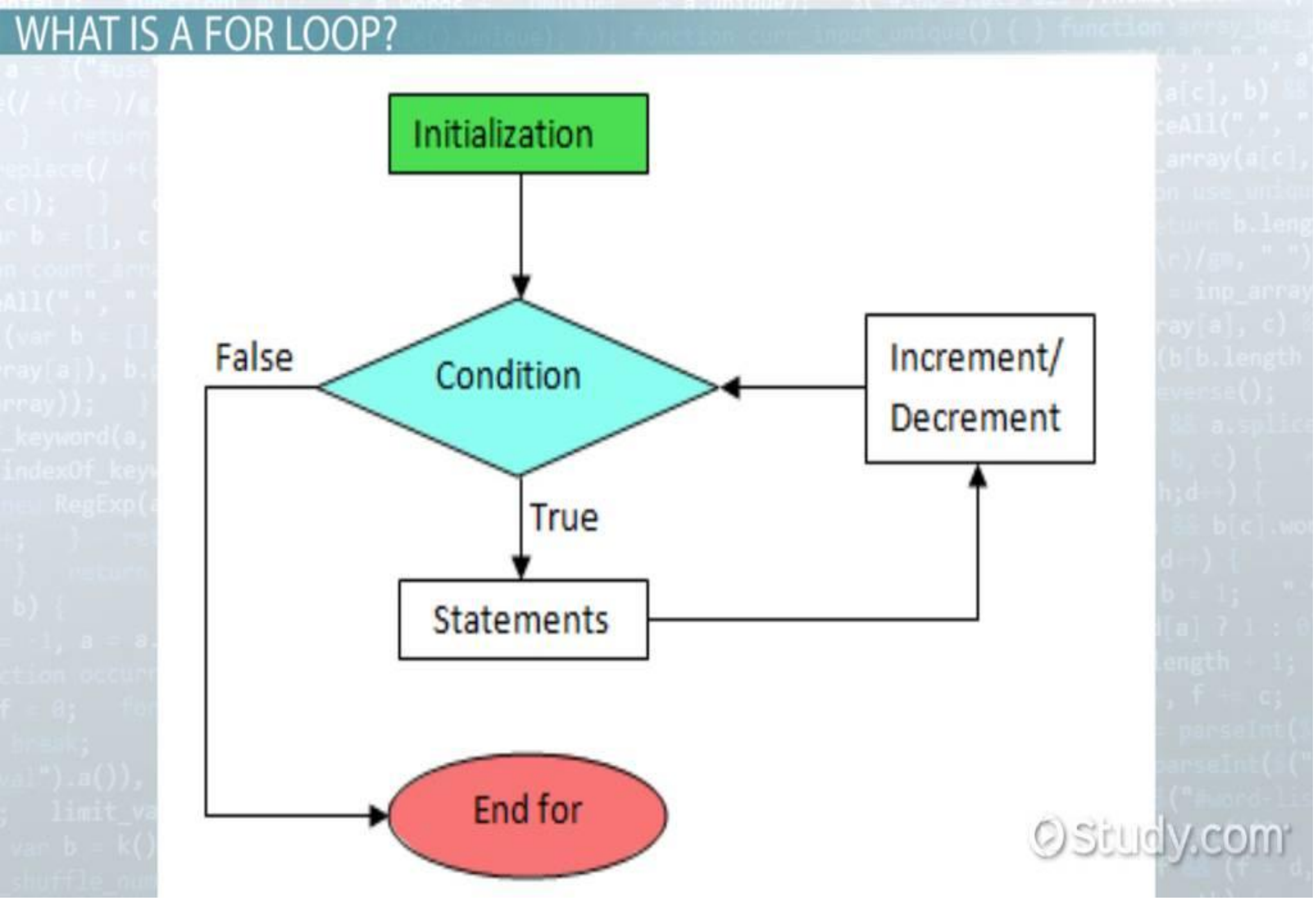
```
        {
         //the body of the loop
         }
```

FOR LOOP

For loop execute the statement of program several times repeatedly until a given condition returns false. It is a count controlled loop in the sense that the program knows in advance how many times the loop is to be executed.

Syntax:-

```
for(initialization; condition; incr/decr)
{
 -------;
 -------; statement
 -------;
 }
```



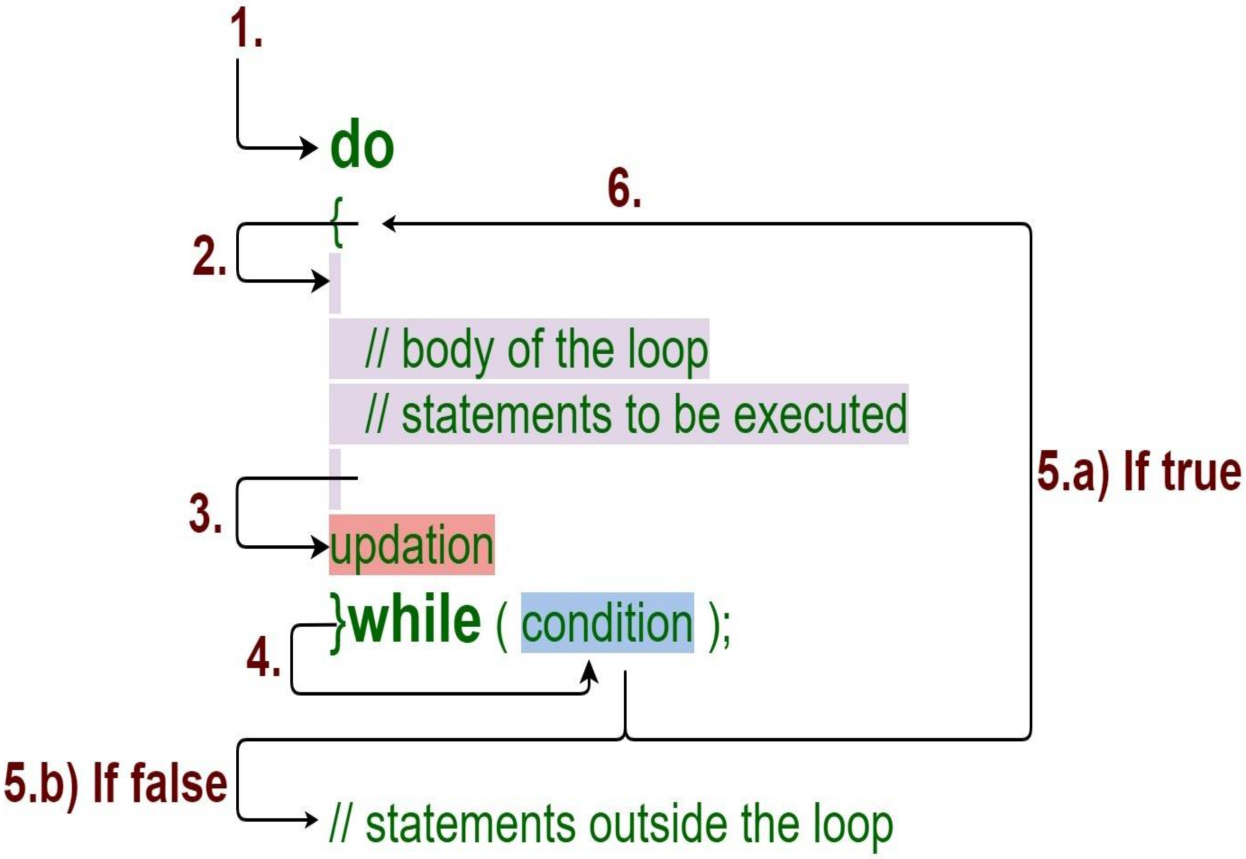
WHAT IS A FOR LOOP?

DO-WHILE LOOP

- Do-while loop is another repetitive control structure provide by c++. The do-while loop is similar to the while loop with one important. The body of do while loop is executed at least once. Only then, the test expression is evaluated.
- Syntax :-

```
do
 {
  -------;
  }
  while();
```

Do - While Loop



1.

```
do
{

  // body of the loop
  // statements to be executed

updation
}while ( condition );
```

6.

2.

3.

4.

5.a) If true

5.b) If false

// statements outside the loop

Thank You!
alt
Ç