Infinite Loop

Learn about infinite loops in C++.

Introduction#

Sometimes, erroneously, we end up writing a piece of code in which a loop condition never evaluates to false and the loop block keeps executing repeatedly. Such types of loops are known as **infinite loops**.

---

*The **infinite loop** keeps executing repeatedly and never terminates.*

Syntax

Let's go over the syntax of the infinite `for` loop.

Keyword

```
for ( ; ; ) {
    statement1;
    statement2;
        .
        .
        .
    statement N;

}
```
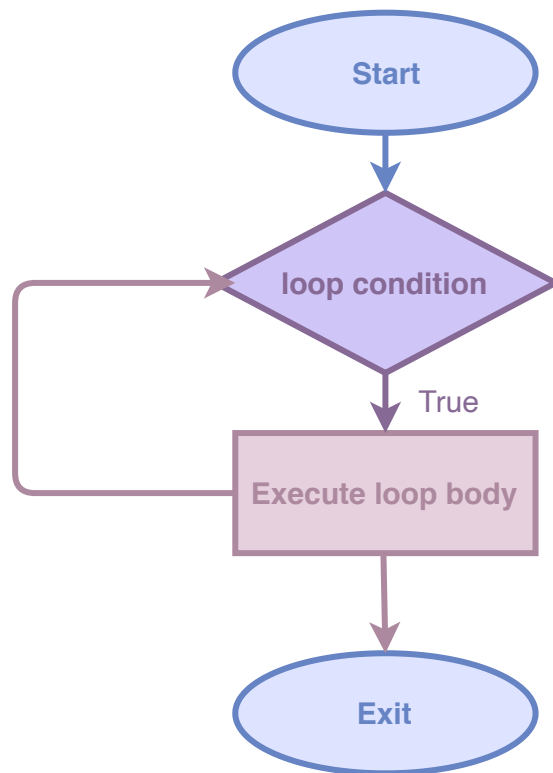
for loop body

We have not specified the termination condition in the `for` loop. Therefore, the `for` loop assumes that the condition is `true` and keeps executing the loop body.

Flowchart#

Let's look at the flowchart of the `for` loop.

- The loop first evaluates the given condition.

- The loop condition always evaluates to `true`. Therefore, it keeps running the code inside the loop body.

Example program#

Press the **RUN** button and see the output!

> ❌ The code given below generates an error.

```
1   #include <iostream>
2
3   using namespace std;
4
5   int main() {
6     for ( ; ;){
7       cout << "Hey, I am infinite loop" << endl;
8     }
9
```

```
10    return 0;
11  }
```

We have not specified the termination condition. Therefore, the loop runs repeatedly and prints `Hey, I am infinite loop` to the console.

On our platform, the compiler stops the execution after `30` seconds and gives you an error if something like that happens. Therefore, you cannot see the output.

However, if you run the same program on your computer, it keeps printing the output to the console and never stops.

> 📝 To stop the infinite loop on your computer, press **CTRL+C**.

Example program#

We can also generate an infinite loop by setting the conditions in such a way that the program never returns `false`.

The program given below will generate an infinite loop!

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5     int number = 1;
6     while (number > 0){
7        cout << number << endl;
8     }
9     return 0;
```

```
10  }
```

In the above code, the condition `number > 0` always returns `true` and thus loop never ends.

Quiz

✅ Q   If there is no condition specified in a `for` loop, it is assumed to be false.

○ **A)** True

Your Answer

✅ **B)** False

Submit Answer

Reset Quiz ↻

That's all about infinite loops. Let's look at nested loops in C++.