For Loop

This lesson will teach the concept and implementation of for loops and nested for loops in C++

We'll cover the following

For Loop: Syntax

The `for` loop is a loop that lets a programmer control exactly how many times a loop will *iterate*.

The *syntax* is as follows:

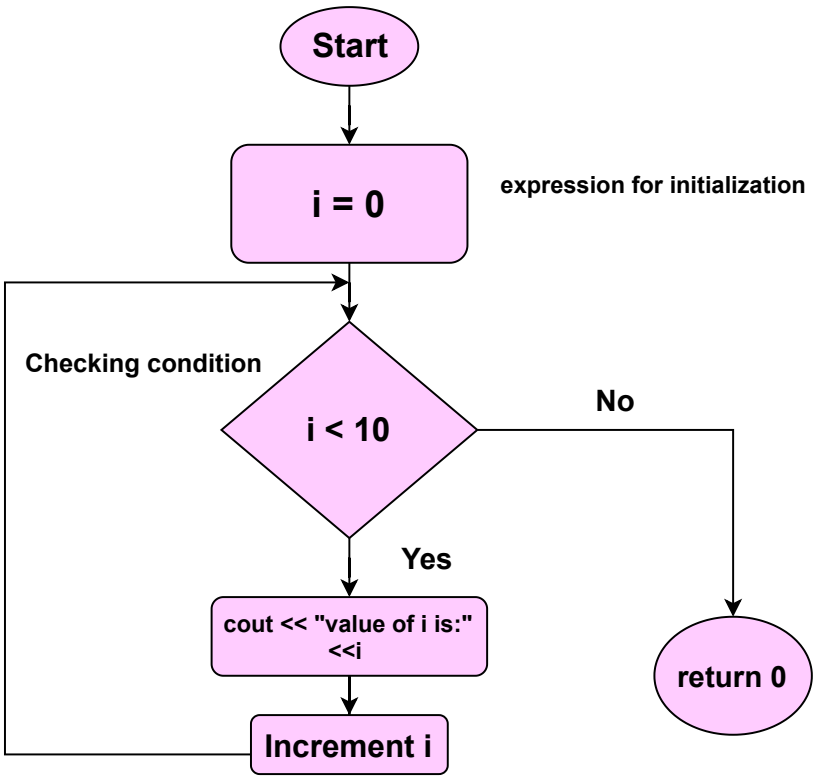
```
for (expression for initialization ; expression for testing ; expression for updating) {
                //body
}
```

Here is an example of how the `for` loop works:

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

int main() {
  for ( int i = 0; i < 10; ++i ) {      // for loop iterates 10 times
    cout << "value of i is: " <<i << endl;
}
  return 0;
}
```

Take a loot at the illustration below to understand the code above more clearly.



Flow Chart for For Loop

The `for` loop code above and the `while` loop below are more or less equivalent.

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

int main() {
  int i = 0;
  while ( i < 10 ) {      // while loop runs till i is less then 10 just like in for loop
    cout << "value of i is: " <<i << endl; //prints value of i
    i++;
}
  return 0;
}
```

What does the for loop do?

- Prior to the *first* iteration, it sets the value of `i` to **0**.
- Next, it tests (like a normal `while` loop) if `i` is *less* than **10**.
- If the statement returns `true`, the body of the loop is run and the program will *print* the value returned by the simple arithmetic statement `i`.

- Next, the terminal cursor moves down to the next line.

- After the loop is finished, `i` is incremented (by **1**), as specified in the update statement, and the conditional is tested again.

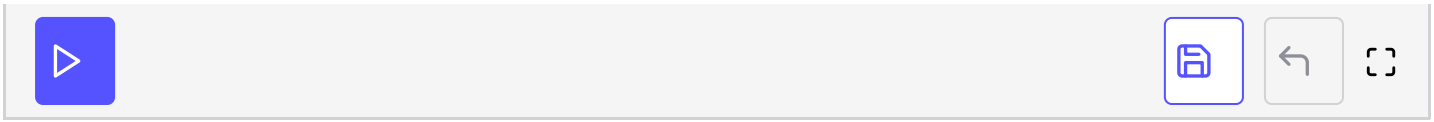
So, this loop will run a total of **10** times, printing the **"i"** value each time. You've just taught your program to count! **Wow!**

The *variable* used in `for` loops is generally an *integer* variable named `i`, `j`, or `k`, and is often initialized prior to the *beginning* of the `for` loop.

Another option is to *initialize* the variable at the **same** time that you declare the variable's *initial* state:

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

int main() {
  for (int i = 0; i < 10; i++) {
    cout << "Value of i+1 is: " <<i+1 << endl;
}
  return 0;
}
```



Nested For Loops

It is possible to *nest* `for` loops. *Nesting* means including one `for` loop in another `for` loop.

The syntax for a **nested** `for` loop is as follows:

```
for (expression for initialization ; expression for testing ; expression for updating ) {
   for (expression for initialization ; expression for testing ; expression for updating) {
      //body
   }
   //body
}
```

Example of Nested For Loop

Let's take a look at an example code to understand *nesting* of `for` loops better.

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

```
int main() {
    int input = 5;


    cout << "How many missiles will you fire?" << endl;
    cout << "I will fire: " << input << " missiles";
    cout << "\n";

    for (int i = 0; i < input; i++) {  // outer for loop
        for (int j = 3; j > 0; j--) {  // inner for loop
            cout << j << " ";
        }
        cout << "Missile " << i+1 << " has launched." << endl;
    }

    cout << "All missiles have been launched." << endl;
    return 0;
}
```

In a nested `for` loop, for a single value of the **outer** loop, in this case, `i`, the inner (*nested*) `for` loop will iterate over all its values, that is, for example for `i=0` the inner (*nested*) loop will run from `j = 3` to `j=1`. After this is done, `i` will be incremented to **1** and the inner loop will again iterate over all its values against this value of `i`. The process continues till all values of `i` are iterated over.

Look at the illustration below which will help you visualize this and help you understand this concept more clearly.

main

input = 5

main

input = 5

How many missiles will you fire?

main

input = 5

How many missiles will you fire?
I will fire: 5 missiles

main

input = 5

How many missiles will you fire?
I will fire: 5 missiles

main

input = 5
i = ?

How many missiles will you fire?
I will fire: 5 missiles

main

input = 5
i = 0
j = ?

How many missiles will you fire?
I will fire: 5 missiles

main

input = 5
i = 0
j = 3

How many missiles will you fire?
I will fire: 5 missiles

main

input = 5
i = 0
j = 3

How many missiles will you fire?
I will fire: 5 missiles
3

main

input = 5
i = 0
j = 2

How many missiles will you fire?
I will fire: 5 missiles
3

main

input = 5
i = 0
j = 2

How many missiles will you fire?
I will fire: 5 missiles
3 2

main

input = 5
i = 0
j = 1

How many missiles will you fire?
I will fire: 5 missiles

3 2

main

input = 5
i = 0
j = 1

How many missiles will you fire?
I will fire: 5 missiles

3 2 1

main

input = 5
i = 0

How many missiles will you fire?
I will fire: 5 missiles

3 2 1

main

input = 5
i = 0

How many missiles will you fire?
I will fire: 5 missiles

3 2 1 Missile 1 has launched

main

input = 5
i = 1

How many missiles will you fire?
I will fire: 5 missiles
3 2 1 Missile 1 has launched

main

input = 5
i = 1
j = 3

How many missiles will you fire?
I will fire: 5 missiles
3 2 1 Missile 1 has launched

main

input = 5
i = 1
j = 3

How many missiles will you fire?
I will fire: 5 missiles

3 2 1 Missile 1 has launched

3

main

input = 5
i = 1
j = 2

How many missiles will you fire?
I will fire: 5 missiles

3 2 1 Missile 1 has launched

3

main

input = 5
i = 1
j = 2

How many missiles will you fire?
I will fire: 5 missiles

3 2 1 Missile 1 has launched

3 2

main

input = 5
i = 1
j = 1

How many missiles will you fire?
I will fire: 5 missiles

3 2 1 Missile 1 has launched

3 2

main

input = 5
i = 1
j = 1

How many missiles will you fire?
I will fire: 5 missiles

3 2 1 Missile 1 has launched

3 2 1

main

input = 5
i = 1

How many missiles will you fire?
I will fire: 5 missiles

3 2 1 Missile 1 has launched

3 2 1

main

input = 5
i = 1

How many missiles will you fire?
I will fire: 5 missiles

3 2 1 Missile 1 has launched

3 2 1 Missile 2 has launched

main

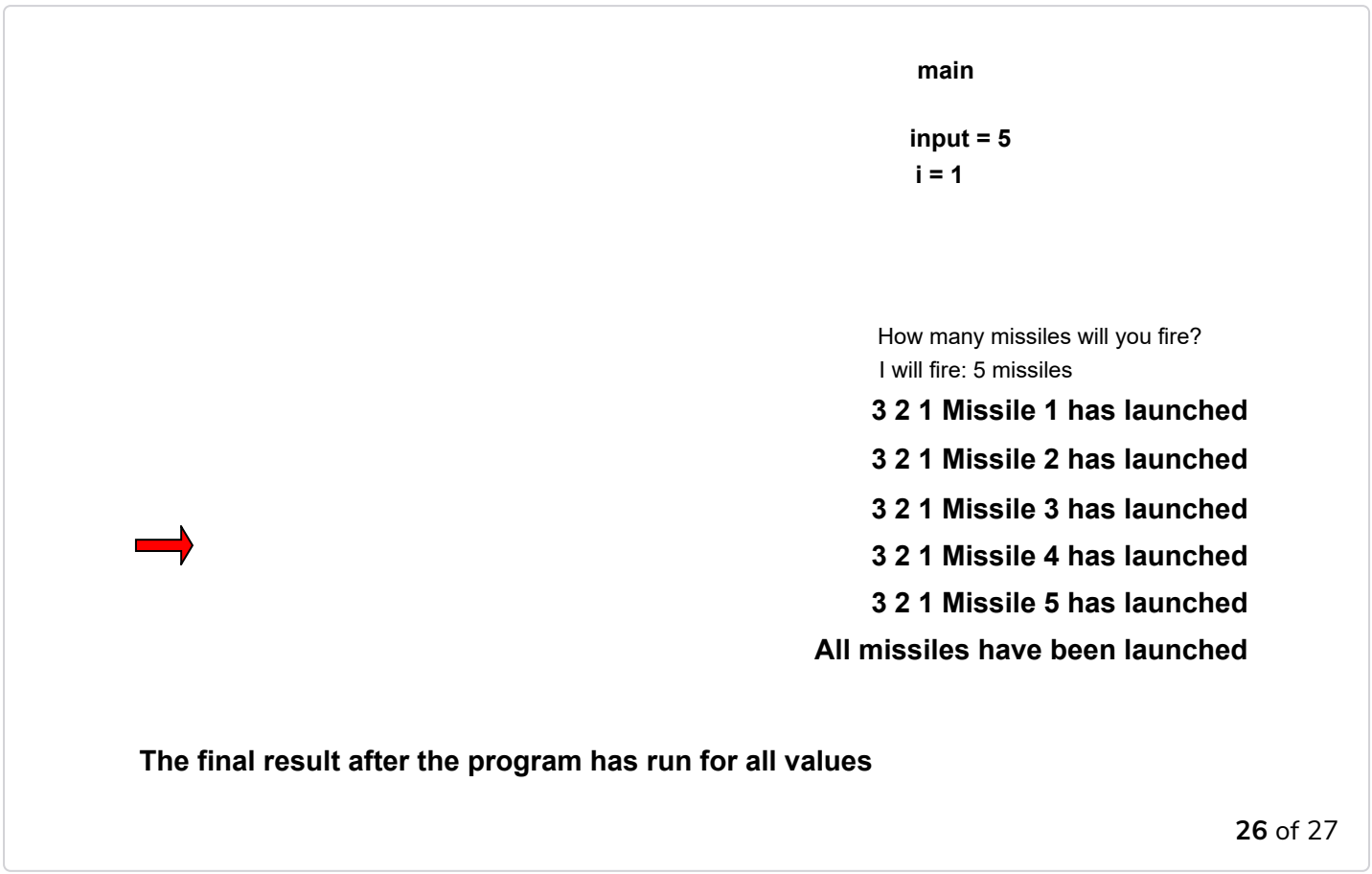
input = 5
i = 1

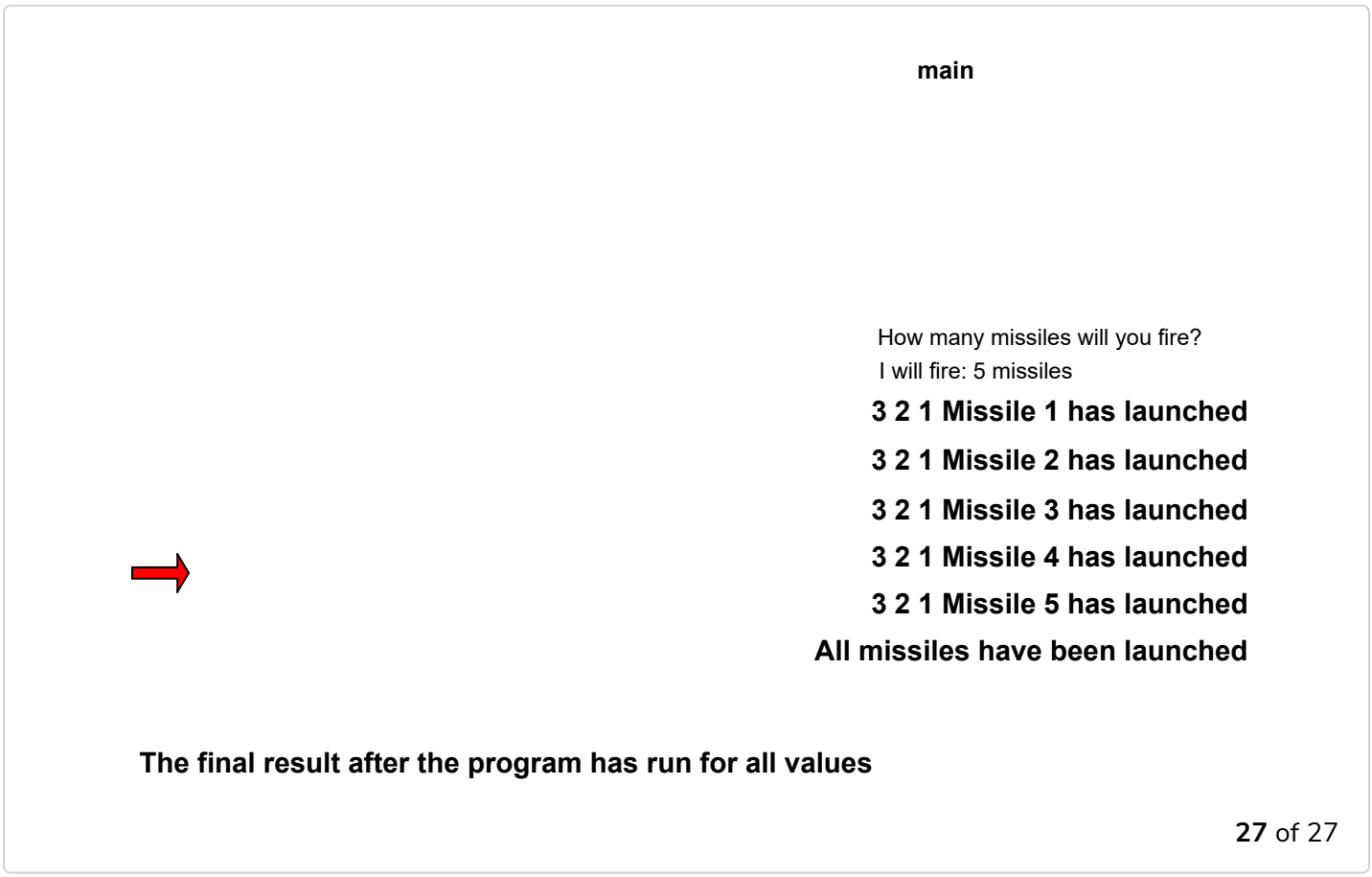
How many missiles will you fire?
I will fire: 5 missiles

3 2 1 Missile 1 has launched

3 2 1 Missile 2 has launched

These iterations will continue for all values of i up till 4





Very interesting right? Now that the concept of `for` loops and *nested* `for` loops is clear let's look at some other interesting stuff related to loops in the next lesson.