1. Learning Objectives

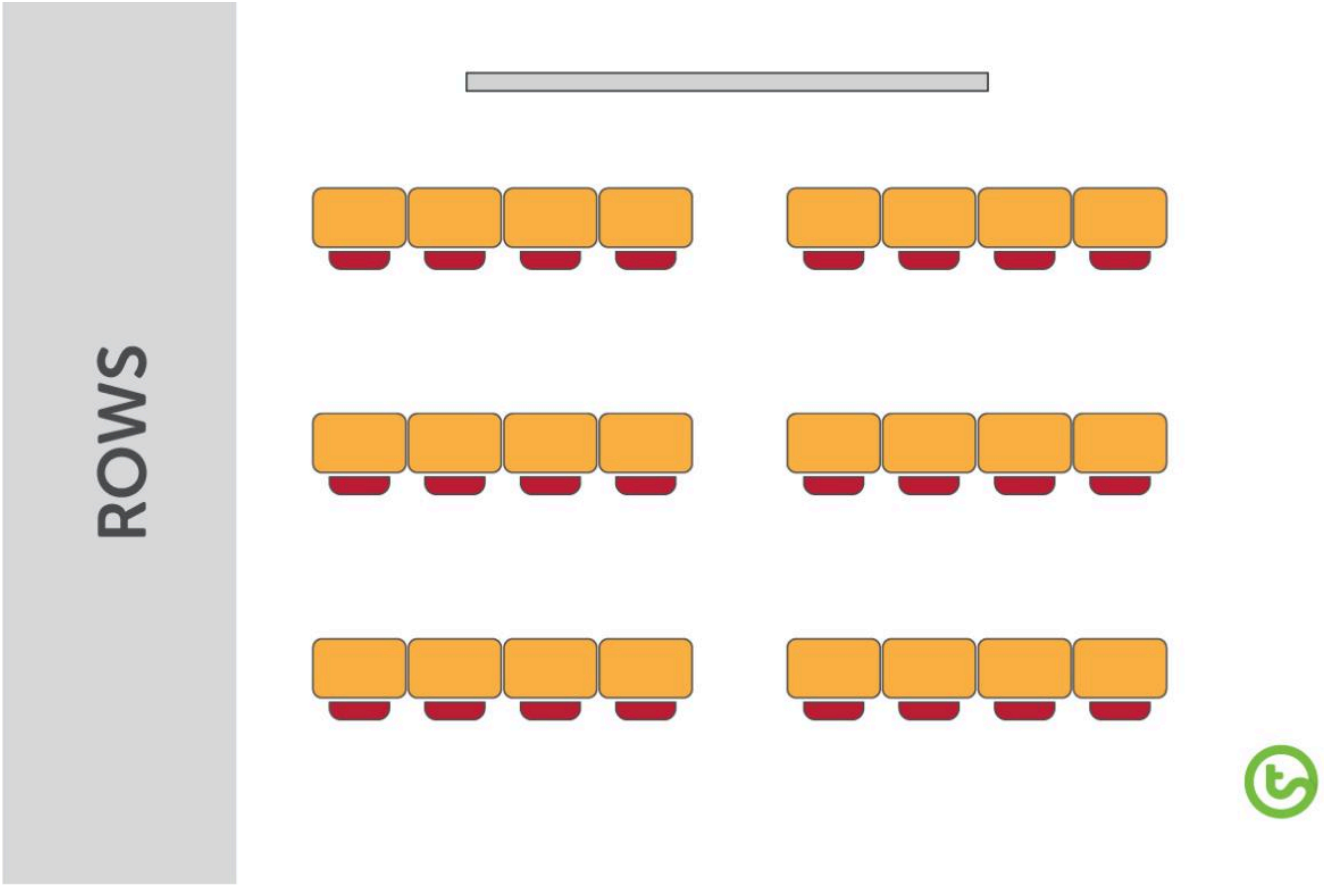
By the end of this lecture, students will be able to:

- Understand nested loops.
- Explain the relationship between outer and inner loops.
- Trace nested loops step by step.
- Create star and number patterns.
- Use `break` to terminate a loop.
- Use `continue` to skip an iteration.

2. What Is a Nested Loop?

A nested loop is a loop placed inside another loop.

Think of a classroom with 3 rows of students, with 4 students in each row. The outer loop represents the rows, while the inner loop represents the students in each row.



teachstarter.com

The outer loop runs once for each row. For every outer-loop iteration, the inner loop completes all its iterations.

3. Example 1: Nested `for` Loops – Printing a Rectangle Pattern

A nested loop is a loop placed inside another loop.

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

int main()
{
    for (int i = 1; i <= 4; i++)
    {
        for (int j = 1; j <= 5; j++)
        {
            cout << "* ";
        }

        cout << endl;
    }

    return 0;
}
```

Output:

```
* * * * *
* * * * *
* * * * *
* * * * *
```

Code Explanation: Read the complete solution with algorithm, pseudocode, dry run, and line-by-line explanation →

Understand the Code

- The outer loop controls the rows. It runs 4 times.
- The inner loop controls the stars in each row. It prints 5 stars.
- After the inner loop finishes, `cout << endl;` starts a new row.
- The inner loop completes all five repetitions for every single outer-loop repetition.

Modify the Code

Try changing the program to produce these patterns.

Pattern A — Rectangle of numbers

```
1 2 3 4
1 2 3 4
1 2 3 4
```

4. Example 2: Right-Angle Triangle of Stars

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

int main()
{
    for (int i = 1; i <= 5; i++)
    {
        for (int j = 1; j <= i; j++)
        {
            cout << "* ";
        }

        cout << endl;
    }

    return 0;
}
```

Output:

```
*
* *
* * *
* * * *
* * * * *
```

Understand the Code

The key statement is:

```cpp
for (int j = 1; j <= i; j++)
```

Unlike the previous example, the inner loop depends on the outer loop's counter `i`.

- When `i = 1`, print 1 star.
- When `i = 2`, print 2 stars.
- When `i = 3`, print 3 stars.
- Continue until 5 rows are printed.

Modify the Code

1. Print a triangle with 7 rows.
2. Replace `*` with `#`.
3. Print numbers instead of stars, as shown below.

```
1
1 2
1 2 3
```

```
1 2 3 4
1 2 3 4 5
```

5. Example 3: Number Pattern

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

int main()
{
    for (int i = 1; i <= 4; i++)
    {
        for (int j = 1; j <= i; j++)
        {
            cout << i << " ";
        }

        cout << endl;
    }

    return 0;
}
```

Output:

```
1
2 2
3 3 3
4 4 4 4
```

Understand the Code

Notice the difference between `i` and `j`:

- `i` determines the row number and the value printed.
- `j` determines how many times that value is printed.

Modify the Code

Produce this pattern:

```
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
```

Hint: Keep the inner loop's condition independent of `i`, so every row contains four numbers.

6. Example 4: Multiplication Tables Using Nested Loops

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

int main()
{
    for (int i = 1; i <= 3; i++)
    {
        cout << "Table of " << i << endl;

        for (int j = 1; j <= 5; j++)
        {
            cout << i << " x " << j
                 << " = " << i * j << endl;
        }

        cout << endl;
    }

    return 0;
}
```

Sample output:

```
Table of 1
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5

Table of 2
2 x 1 = 2
2 x 2 = 4
...
```

Understand the Code

The outer loop selects the table. The inner loop calculates five multiplication results for each table.

Modify the Code

1. Display tables from 1 to 10.
2. Display each table up to 10.
3. Ask the user how many tables to display.

7. Practice Coding Exercises

Solve Coding Exercises 29–31 in the `Nested Loop` section. Write, run, and test each C++ program to verify the output.