Table of contents

Press Escape key to exit fullscreen.

01

Selection Structure

Ternary statement,
Switch case

Ternary Statement

The ternary operator, also known as the **conditional operator**, is a **shorthand way of writing an if-else statement**.

It allows you to write a concise **conditional expression** in a **single line**.

Ternary operator is denoted by **?:**

Syntax of Ternary Statement

```
(condition) ? expression_if_true : expression_if_false;
```

Example of Ternary Statement

```
(a > b) ? cout << “a is max” : cout << “b is max”;
```

OR

```
(a > b)
    ? cout << “a is max”
    : cout << “b is max”;
```

Switch Statement

The switch statement is a control flow statement that **allows a variable to be tested for equality against a list of values**.

Note: Switch statement mainly used for **comparison purpose** and for creating **menu-driven programs**.

Syntax of Switch Statement

```
switch (expression) {
    case value1:
        // code to be executed if expression matches value 1
        break;
    case value2:
        // code to be executed if expression matches value 2
        break;
    // additional cases as needed
    default:
        // code to be executed if expression doesn't match any case
}
```

Switch vs Ladder Statement

```
switch (grade) {
    case ‘A’:
        cout << “Excellent”;
        break;
    case ‘B’:
        cout << “Average”;
        break;
    default:
        cout << “Failed”;
}
```

VS

```
if (grade == ‘A’) {
      cout << “Excellent”;
}
else if (grade == ‘B’) {
      cout << “Average”;
}
else {
     cout << “Failed”;
}
```

TL;DR

Ternary Statement

- It allows you to write a concise conditional expression in a single line.
- A shorthand way of writing an if-else statement.
- Ternary operator is denoted by ?:

Switch Statement

- It allows a variable to be tested for equality against a list of values.
- Mainly used for comparison purpose and for creating menu-driven programs.
- switch, case, break, and default keywords are used.