Decision Making

Terms

Boolean expression
Comparison operators
Conditional operator
If statement
Logical operators
Nesting if statements
Switch statement

Summary

- We use *comparison operators* to compare values.
- A *Boolean expression* is a piece of code that produces a Boolean value.
- With *Logical operators* we can combine Boolean expressions and represent more complex conditions.
- With the *logical AND operator* (&&) both operands should be true. If either of them is false, the result will be false.
- With the *logical OR operator* (||), the result is true if either of the operands is true.
- The *logical NOT operator* (!) reverses a Boolean value.
- Using ***if*** and ***switch*** *statements*, we can control the logic of our programs.
- An if statement can have zero or more **else if** clauses for evaluating additional conditions.

- An if statement can optionally have an **else** clause.
- We can code an if statement within another. This is called *nesting* if statements.
- Using the *conditional operator* we can simplify many of the if/else statements.
- We use **switch** statements to compare a variable against different values.
- A switch block often has two or more *case labels* and optionally a *default label*.
- Case labels should be terminated with a **break** statement; otherwise, the control moves to the following case label.
- Switch statements are not as flexible as if statements but sometimes they can make our code easier to read.


```
// Comparison operators
bool a = 10 > 5;
bool b = 10 == 10;
bool c = 10 != 5;


// Logical operators
bool d = a && b; // Logical AND
bool e = a || b; // Logical OR
bool f = !a;     // Logical NOT


if (temperature < 60) {
    // ...
}
else if (temperature < 90) {
    // ...
}
else {
    // ...
}


// Conditional operator
double commission = (sales < 10'000) ? .05 : .1;
```



```
switch (menu) {
    case 1:
        // ...
        break;
    case 2:
        // ...
        break;
    default:
        // ...
}
```

