

MODULE PRACTICE

if Statement

```
if (a == 10) {  
    // Code goes here  
}
```

An `if` statement is used to test an expression for truth.

- If the condition evaluates to `true`, then the code within the block is executed; otherwise, it will be skipped.

Logical Operators

```
if (coffee > 0 && donut > 1) {  
    // Code runs if both are true  
}  
  
if (coffee > 0 || donut > 1) {  
    // Code runs if either is true  
}  
  
if (!tired) {  
    // Code runs if tired is false  
}
```

Logical operators can be used to combine two different conditions.

- `&&` requires both to be true (`and`)
- `||` requires either to be true (`or`)
- `!` negates the result (`not`)

Relational Operators

```
if (a > 10) {  
    // 🙌 means greater than  
}
```

Relational operators are used to compare two values and return `true` or `false` depending on the comparison:

- `==` equal to
- `!=` not equal to
- `>` greater than
- `<` less than
- `>=` greater than or equal to
- `<=` less than or equal to

`else if` Statement

```
if (apple > 8) {  
    // Some code here  
}  
else if (apple > 6) {  
    // Some code here  
}  
else {  
    // Some code here  
}
```

One or more `else if` statements can be added in between the `if` and `else` to provide additional condition(s) to check.

else Clause

```
if (year == 1991) {  
    // This runs if it is true  
}  
else {  
    // This runs if it is false  
}
```

An `else` clause can be added to an `if` statement.

- If the condition evaluates to `true`, code in the `if` part is executed.
- If the condition evaluates to `false`, code in the `else` part is executed.

`switch` Statement

```
switch (grade) {  
    case 9:  
        std::cout << "Freshman\n";  
        break;  
    case 10:  
        std::cout << "Sophomore\n";  
        break;  
    case 11:  
        std::cout << "Junior\n";  
        break;  
    case 12:  
        std::cout << "Senior\n";  
        break;  
    default:  
        std::cout << "Invalid\n";  
        break;  
}
```

A `switch` statement provides a means of checking an expression against various `case` s. If there is a match, the code within starts to execute. The `break` keyword can be used to terminate a case.

`default` is executed when no case matches.