

MODULE PRACTICE

else Statements

An `else` statement is accessed when all preceding `if` and/or `else-if` statements return `false`.

```
if (x > 3) {  
    printf("x is greater than 3");  
} else if (x < 3) {  
    printf("x is less than 3");  
} else {  
    printf("x equals 3");  
}
```

Types of Conditionals

A conditional in C can be written using `if`, `else-if`, `else`, ternary operators, and `switch` statements.

switch Statements

A `switch` statement is a condensed series of cascading `else` statements. It tests a value and compares it against multiple cases.

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

else-if Statements

An `else-if` statement tests an expression and must come after an existing `if` or `else-if`.

```
if (x > 3) {
    printf("x is greater than 3");
} else if (x < 3) {
    printf("x is less than 3");
}
```

Dangling else Statement

A dangling `else` statement results when it's ambiguous which conditional the `else` statement is attached to.

Operators and Conditionals

A conditional in C can use logical operators such as `&&` and `||` to test multiple expressions and `!` to negate an expression.

if Statements

An `if` statement tests an expression and executes code based on its truth.

```
if (x == 3) {  
    printf("x is 3!");  
}
```

Ternary Operators

A ternary operator is a condensed `if-else` statement.

```
min = a < b ? a : b; // This is the same as the if-else below
```

```
if (a < b) {  
    min = a;  
} else {  
    min = b;  
}
```