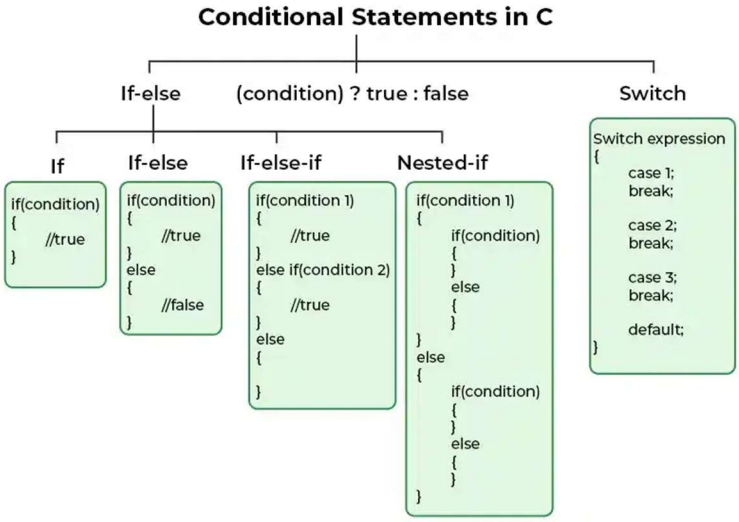
IV. Control Structures

Conditional Statements

Conditional statements, also known as control structures, are essential components of programming that allow you to make decisions in your code based on certain conditions. These statements enable you to execute different **blocks of code** depending on whether a specified **condition is true or false**. In this detailed explanation, we'll explore conditional statements in depth, including their types, syntax, and usage.



There are **three main types** of conditional statements in most programming languages:

1. **if Statements:**
    - The **if** statement is the most basic form of a conditional statement.
    - It allows you to execute a block of code only if a specified condition is true.

    Syntax:

    ```
    if (condition) {
        // Code to execute if the condition is true
    }
    ```

    Example:

    ```
    int age = 25;
    if (age >= 18) {
        printf("You are an adult.");
    }
    ```

In this example, the message "**You are an adult**" will be printed only if the **age** is greater than or equal to 18.

2. **if-else Statements:**

   - The **if-else** statement extends the **if** statement by allowing you to specify an alternative block of code to execute if the condition is false.

Syntax:

```
if (condition) {
   // Code to execute if the condition is true
} else {
   // Code to execute if the condition is false
}
```

Example:

```
int score = 75;
if (score >= 60) {
   printf("You passed the exam.");
} else {
   printf("You failed the exam.");
}
```

In this example, the appropriate message is printed based on whether the **score** is greater than or equal to 60.

3. **if-else if-else Statements:**

   - The **if-else if-else** statement allows you to test multiple conditions in a sequence and execute the code block associated with the first true condition.

Syntax:

```
if (condition1) {
   // Code to execute if condition1 is true
} else if (condition2) {
   // Code to execute if condition2 is true
} else {
   // Code to execute if none of the conditions are true
```

```
}
```

Example:

```
int number = 7;
if (number < 0) {
   printf("The number is negative.");
} else if (number == 0) {
   printf("The number is zero.");
} else {
   printf("The number is positive.");
}
```

In this example, the code will print one of the three messages depending on the value of the **number**.

Nested Conditional Statements:

You can also nest conditional statements, which means placing one conditional statement inside another. This allows for more complex decision-making.

Example:

```
int age = 25;
if (age >= 18) {
   if (age < 30) {
      printf("You are a young adult.");
   } else {
      printf("You are an adult.");
   }
} else {
   printf("You are a minor.");
}
```

In this example, there are nested if statements to categorize individuals into different age groups.

Switch Statements:

The **switch** statement provides a way to choose between many code blocks based on the value of an expression. It's often used when you have a single variable to compare against multiple possible values.

Syntax:

```
switch (expression) {
   case value1:
      // Code to execute if expression equals value1
      break;
   case value2:
      // Code to execute if expression equals value2
      break;
   // ... other cases ...
   default:
      // Code to execute if expression doesn't match any case
}
```

Example:

```
int day = 3;
switch (day) {
   case 1:
      printf("Monday");
      break;
   case 2:
      printf("Tuesday");
      break;
   case 3:
      printf("Wednesday");
      break;
   default:
      printf("Other day");
}
```

In this example, the code will print "Wednesday" because the value of **day** is 3.

Conditional Ternary Operator:

The conditional ternary operator **? :** is a shorthand way to write simple conditional expressions in a single line.

Syntax:

```
condition ? expression_if_true : expression_if_false;
```

Example:

```
int score = 75;
char result = (score >= 60) ? 'P' : 'F';
```

In this example, the value of **result** will be 'P' if the **score** is greater than or equal to 60, and 'F' otherwise.

Conditional statements are a powerful feature of programming that enables your code to make decisions based on certain conditions. They are essential for creating flexible and responsive programs.