C Comments

Comments are pieces of text within a program that are ignored by the compiler. They exist solely for documentation purposes and are immensely useful in software development.

Why Are Comments Important?

Comments are important because of the following reasons:

- **Code Clarity:** Comments help you understand the logic and functionality provided by specific parts of the code.
- **Collaboration**: In a software company, codebases often span multiple files and are written by various developers. Comments make it easier for team members to navigate and comprehend the code.
- **Self-Reference**: Even when revisiting your own code after weeks or months, comments help recall what you were thinking at the time of writing.

Example

C

```c
/* This is an example of
multiline comment */
#include <stdio.h>
int main() {

    // Example of single line comment
    printf("Hello GeeksforGeeks -- with comments");
    return 0;
}
```

```
Hello GeeksforGeeks -- with comments
```

Output

```
GeeksforGeeks
```

In the above code, we have written two comments in two different ways:

Types of Comments in C

C programming language supports two types of comments:

1. Single-line Comments

Single-line comments start with // and continue to the end of the line. Historically, these were called "C++ style comments" because they were introduced in C++ and later added to the C language.

2. Multi-line Comments

Multi-line comments begin with **/*** and end with ***/.** They are typically used for documenting larger sections of code. If the ending marker is omitted, the compiler treats everything following the start marker as part of the comment.

Multi-line comments were present in the original version of C. Multi-line comments can also be used for single lines, and single-line comments can be used in consecutive lines to create a multi-line comment.

C

```c
#include <stdio.h>

/* Single-line comment using multi-line syntax */
int main(){
    printf("The comments aren't seen");
}
// This is a multi-line comment
// written using single-line syntax.
```

```
The comments aren't seen
```

Typically, multi-line comments are preferred for longer explanations, while single-line comments are better suited for short notes or clarifications.

Real-World Use Cases

Let's take a look at some examples of real-world usage of comments:

1. Describing Functions

Comments are often used to describe the functionality of a function. For example:

C

```c
/* Removes special characters, leading
and trailing spaces from the inputStr. */
```

```
void processInput(char inputStr[])
{
    . . . . .
    . . . . .
    . . . . .
}
```

2. Top-Level Comments in Files

In the software industry, it is common to include a few lines of comments at the top of each file to explain:

- The purpose of the file.
- The functions it contains.
- Why the file exists and how it is used.

3. Modifying Standard Algorithms

When modifying standard algorithms, comments indicate the changes and their purpose. For example:

> // Modifying the standard binary search to count occurrences instead of finding a single match.

Avoid Unnecessary Comments

While comments are extremely useful, it’s important not to include unnecessary ones. For instance:

C

```
/* Removes special characters, leading
and trailing spaces from the inputStr. */
void processInput(char inputStr[])  // This is a void function
{
    . . . . .
    . . . . .
    . . . . .
}
```

The comment "This is a void function" is not helpful because the declaration already conveys that the function is of type void.