In C programming, comments are used to add explanatory notes or documentation to the source code. Comments are ignored by the compiler and are not executed as part of the program.

There are two types of comments in C programming:

1. Single-line comments:

These comments begin with // and extend to the end of the line. Anything after // on the same line is considered a comment and is ignored by the compiler.

Example:

```
// This is a single-line comment

printf("Hello, world!"); // This line prints "Hello, world!"
```

2. Multi-line comments:

These comments begin with /* and end with */. Anything between these symbols is considered a comment and is ignored by the compiler. Multi-line comments can span multiple lines.

Example:

```
/* This is a multi-line comment
```

```
    that spans multiple lines */

printf("Hello, world!"); /* This line prints "Hello, world!" */
```

Comments are useful for documenting the purpose of variables, functions, and code blocks, as well as for debugging and troubleshooting. Good use of comments can help make the code more readable and understandable to other developers who may need to work with it in the future.

C Programming most asked questions on comments:

Q: What is a comment in C programming?

A: A comment in C programming is a piece of code that is ignored by the compiler and is used to add explanatory notes or documentation to the source code. Comments are not executed as part of the program.

Q: What are the two types of comments in C programming?

A: The two types of comments in C programming are single-line comments and multi-line comments.

Q: How do you write a single-line comment in C programming?

A: A single-line comment in C programming begins with // and extends to the end of the line. Anything after // on the same line is considered a comment and is ignored by the compiler.

Q: How do you write a multi-line comment in C programming?

A: A multi-line comment in C programming begins with /* and ends with */. Anything between these symbols is considered a comment and is ignored by the compiler. Multi-line comments can span multiple lines.

Q: Why are comments important in programming?

A: Comments are important in programming because they help make the code more readable and understandable to other developers who may need to work with it in the future. Comments can also be used to document the purpose of variables, functions, and code blocks, as well as for debugging and troubleshooting.

Q: Can comments be nested in C programming?

A: Yes, comments can be nested in C programming. However, it is not recommended to nest comments too deeply, as it can make the code harder to read and understand.

Q: What is the purpose of using comments in C programming?

A: The purpose of using comments in C programming is to add explanatory notes or documentation to the source code. Comments help make the code more readable and understandable to other developers who may need to work with it in the future, and can be used to document the purpose of variables, functions, and code blocks, as well as for debugging and troubleshooting.

MCQs on Comments in C Programming:

Which of the following is the symbol used for single-line comments in C programming?

A. /

B. //

C. #

D. ;

Answer: B. //

Which of the following is the symbol used for multi-line comments in C programming?

A. /

B. //

C. #

D. /* /

Answer: D. / */

What is the purpose of comments in C programming?

A. To add functionality to the code

B. To change the output of the program

C. To add explanatory notes or documentation to the code

D. To speed up the program

Answer: C. To add explanatory notes or documentation to the code

Can comments be nested in C programming?

A. Yes, but it is not recommended

B. No, comments cannot be nested

C. Yes, and it is recommended to nest comments as much as possible

D. Only single-line comments can be nested
Answer: A. Yes, but it is not recommended

Which of the following is a benefit of using comments in C programming?
A. Making the code run faster
B. Making the code shorter
C. Making the code more difficult to read
D. Making the code more readable and understandable to other developers
Answer: D. Making the code more readable and understandable to other developers

Related posts:

1. C prgoram to convert inch to feet
2. C program to convert KM to CM
3. C program to convert meter to centimeter
4. C program to calculate remainder, difference, division, product
5. C program to use printf() without semicolon ” ; “
6. C program to swap two numbers using 2 variables
7. C program to find nth term using Arithmetic progrssion
8. C program to find sum of first n even positive numbers
9. C program to calculate sum of first n even numbers
10. C program to find nth odd number
11. C program to find sum of first n odd positive numbers
12. C program to calculate perimeter and area of a rectangle
13. C program to calculate perimeter and area of a square
14. C program to calculate Perimeter and Area of Circle
15. Function in C Programming
16. C Programming Q & A

17. Main function in C Programming Q and A
18. Void main in C Programming
19. Variables Q and A in C Programming
20. Write a C Program to find the percentage of marks ?
21. Write a c program to find age of a person ?
22. Write a c program to get table of a number
23. What is Break statement in C Programming ?
24. Write a c program to generate all combinations of 1, 2 and 3 using for loop.
25. Write a C program to print all the prime numbers between 1 to 50.
26. Write a C program to get factorial of a number ?
27. What is user defined function in C programming ?
28. Difference between C and C++ Programming ?
29. Difference between C, C++ and Java Programming
30. C program addition of numbers using pointer
31. C Syntax
32. Variables in C
33. Data types in C
34. Format specifiers in C
35. Type Conversion in C
36. Constants in C
37. Operators in C
38. Pre and Post Increament Practice Problems
39. Pre and Post Increament
40. Array in C
41. C Introduction
42. C Get Started
43. C Pointers

44. C History
45. C Program Compiling and running
46. C While loop
47. C Do While Loop
48. C For loop
49. break and continue statement
50. Control Statements in C
51. C if-else ladder
52. C if statements
53. C 2-Dimensional array
54. C String library functions
55. C Functions
56. C Functions Categories
57. C Actual Arguments
58. Write a program that prints the message “Hello, World!”
59. Write a program that asks the user to enter two numbers, and then prints the sum of those two numbers.
60. Write a program that asks the user to enter a number and then determines whether the number is even or odd.
61. Write a program that swaps the values of two variables.
62. Write a program that asks the user to enter a number and then calculates and prints its factorial.
63. Write a program that asks the user to enter a number N and then prints the first N numbers in the Fibonacci sequence
64. Write a program that swaps the values of two variables without using a temporary variable
65. Converts a number into integer, float, and string

66. Program to find the length of the string
67. Program to convert string to uppercase or lowercase
68. Program to prints the numbers from 1 to 10.
69. What is identifier expected error
70. Difference between static and non static methods in Java
71. C String Input
72. C Character input
73. C Programming Variables MCQ
74. Object & Classes
75. C Programming find the output MCQs