Table of Contents

Function Definition and Declaration:

- A function in C consists of a function declaration (prototype) and a function definition.
- The function declaration specifies the function name, return type, and parameter types (if any). It informs the compiler about the existence and signature of the function.
- The function definition contains the actual implementation of the function, including the function body (code).
- Here's an example of a function declaration and definition:

C

```c
// Function declaration
int addNumbers(int a, int b);

// Function definition
int addNumbers(int a, int b) {
    int sum = a + b;
    return sum;
}
```

Calling Functions:

- To use a function, you need to call it from another part of your program.
- Function calls typically include the function name followed by parentheses, which may contain arguments (if any).
- The return value of a function can be assigned to a variable or used directly in expressions.
- Here's an example of calling the addNumbers() function:

C

```c
int result = addNumbers(3, 4);
printf("The sum is: %d\n", result);
```

Function Parameters:

- Functions can have parameters, which allow you to pass values to the function.
- Parameters are specified in the function declaration and definition within parentheses.
- The values passed to the function are called arguments.
- Here's an example of a function with parameters:

C

```c
int multiply(int a, int b) {
    int product = a * b;
    return product;
}
```

Function Return Type:

- The return type of a function specifies the type of value the function returns.
- Functions can have various return types, such as int, float, char, void, etc.
- If a function doesn't return a value, the return type is void.
- Here's an example of a function with a return type of float:

C

```c
float calculateAverage(float a, float b, float c) {
    float average = (a + b + c) / 3;
    return average;
}
```

Function Call by Value:

- By default, C uses call by value, which means that function arguments are passed as copies of their values.
- Any changes made to the parameter within the function do not affect the original argument.
- Here's an example illustrating call by value:

C

```c
void increment(int value) {
    value++;
}

int main() {
    int num = 5;
```

```
    increment(num);
    printf("The value is: %d\n", num);  // Output: The value is: 5
    return 0;
}
```

Function Prototypes:

- Function prototypes allow you to declare a function before its actual definition.
- Prototypes specify the function's name, return type, and parameter types.
- Prototypes are typically placed in a header file that can be included in multiple source files.
- This enables functions to be defined later in the code but still be usable in other parts of the program.
- Here's an example of a function prototype:

C

```c
// Function prototype
int multiply(int a, int b);

int main() {
    int result = multiply(3, 4);
    printf("The product is: %d\n", result);
    return 0;
}

// Function definition
int multiply(int a, int b) {
    int product = a * b;
    return product;
```

```
}
```

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. Comments in C
33. Variables in C
34. Data types in C
35. Format specifiers in C
36. Type Conversion in C
37. Constants in C
38. Operators in C
39. Pre and Post Increament Practice Problems
40. Pre and Post Increament
41. Array in C
42. C Introduction
43. C Get Started
44. C Pointers
45. C History
46. C Program Compiling and running
47. C While loop
48. C Do While Loop
49. C For loop
50. break and continue statement

51. Control Statements in C
52. C if-else ladder
53. C if statements
54. C 2-Dimensional array
55. C String library 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