Functions in C can be categorized into several types based on their characteristics and purpose.

Here Are Some Common Categories Of Functions:

1. Standard Library Functions:

- These are functions provided by the C standard library, such as <stdio.h>,<stdlib.h>,<string.h> etc.
- Standard library functions serve various purposes, including input/output operations, memory management, string manipulation, mathematical calculations, and more.
- Examples: printf(), scanf(), malloc(), strlen(), strcmp(), etc.

2. User-Defined Functions:

- User-defined functions are created by the programmer to perform specific tasks within a program.
- These functions provide modularity, code reusability, and abstraction, making the program easier to understand and maintain.
- Examples: Functions that perform calculations, validate input, process data, implement algorithms, etc.

---

3. Recursive Functions:

- Recursive functions are functions that call themselves directly or indirectly.
- They are useful for solving problems that can be divided into smaller sub-problems of the same type.
- Recursive functions have a base case that defines the termination condition, and a recursive case that calls the function with a smaller input.
- Examples: Calculating factorials, computing Fibonacci series, traversing data structures recursively, etc.
- Example of recursive functions:

C

```c
#include <stdio.h>

int sum(int n) {
    // Base case: if n is 1, return 1
```

```
    if (n == 1) {
        return 1;
    }

    // Recursive case: add n to the sum of numbers from 1 to (n-1)
    return n + sum(n - 1);
}

int main() {
    int num = 5;
    int result = sum(num);
    printf("The sum of numbers from 1 to %d is: %d\n", num, result);
    return 0;
}
```
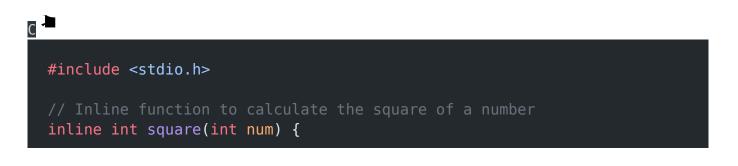
4. Library Functions:

- Library functions are functions grouped together in a library for specific purposes.
- Libraries contain related functions that can be reused across multiple programs.
- These functions provide additional functionality beyond what is available in the standard library.
- Examples: Math library functions (), time-related functions (), graphics functions (), etc.

5. Callback Functions:

- Callback functions are functions that are passed as arguments to other functions.
- The receiving function can call the callback function at a specific point, allowing for customization and extensibility.
- Callback functions are commonly used in event-driven programming or when implementing data structures like sorting algorithms.
- Examples: Event handlers, comparison functions for sorting, callback functions in APIs, etc.

---

6. Inline Functions:

- Inline functions are small functions that are expanded at the call site instead of being called like regular functions.
- The purpose of inline functions is to avoid the overhead of function calls and improve performance.
- Inline functions are defined using the inline keyword.
- Examples: Small utility functions, simple arithmetic operations, etc.
- Example of inline function:

C

```c
#include <stdio.h>

// Inline function to calculate the square of a number
inline int square(int num) {
```

```
    return num * num;
}

int main() {
    int num = 5;
    int result = square(num);
    printf("The square of %d is: %d\n", num, result);
    return 0;
}
```

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

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