Programming with C and C++

CSC-101 (Lecture 17)

Dr. R. Balasubramanian
Professor
Department of Computer Science and Engineering
Mehta Family School of Data Science and Artificial Intelligence
Indian Institute of Technology Roorkee
Roorkee 247 667

bala@cs.iitr.ac.in
https://faculty.iitr.ac.in/cs/bala/

2-D character Array

</> source code

```c
#include <stdio.h>

int main(void) {
    const int ROWS = 3;
    const int COLS = 30;

char grid[3][30] = {
    {"Hello"},
    {"I am in the Dept. of CSE"},
    {"IIT Roorkee"}
};

char searchChar = 'I';
int count = 0;

```

```
15
16  // Searching for the character and counting its occurrences
17  for (int i = 0; i < ROWS; i++) {
18      for (int j = 0; j < COLS; j++) {
19          if (grid[i][j] == searchChar) {
20              count++;
21          }
22      }
23  }
24  printf("Occurrences of '%c ': %d", searchChar,count);
25      return 0;
26  }
27
```

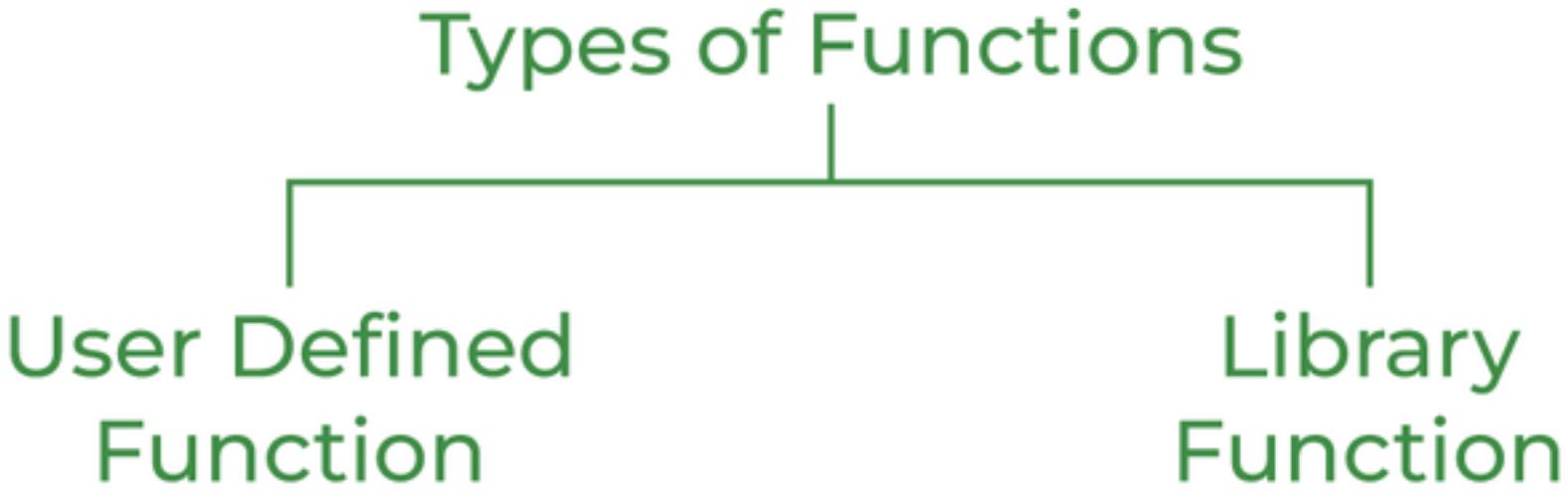
Success #stdin #stdout 0s 5516KB

```
Occurrences of 'I ': 3
```

https://ideone.com/vRXDgv

C Functions

- In C, we can divide a large program into the basic building blocks known as ***function***.
- The function contains the set of programming statements enclosed by { }.
- A function can be called multiple times to provide reusability and modularity to the C program.
- In other words, we can say that the collection of functions creates a program.
- The function is also known as procedure or subroutine or method in other programming languages.


Types of Functions
User Defined Function
Library Function

Function Aspects

SN	C function aspects	Syntax
1	Function declaration/function prototype	`return_type function_name (argument list);`
2	Function call	`function_name (argument_list)`
3	Function definition	`return_type function_name (argument list) {function body;}`

Function Prototype

</> source code

```c
// C program to illustrate the function prototye
#include <stdio.h>

// Function prototype
float calculateRectangleArea(float length, float width);

int main()
{
    float length = 15.0;
    float width = 3.0;

    // Function call
    float area = calculateRectangleArea(length, width);

    printf("The area of the rectangle is: %.2f\n", area);

    return 0;
}

```

```
19
20  // Function definition
21  float calculateRectangleArea(float length, float width)
22  {
23      return length * width;
24  }
25
```

https://ideone.com/qPmD2c

input Output

Success #stdin #stdout 0s 5564KB

```
The area of the rectangle is: 45.00
```

Syntax

```
return_type function_name(parameter_list);
```

```
4   // Function prototype
5   float calculateRectangleArea(float length, float width);
```

Arguments

```
// Function call
float area = calculateRectangleArea(length, width);
```

Function Definition

```c
// C program to illustrate the function definition
#include <stdio.h>

// Function definition
float calculateRectangleArea(float length, float width)
{
    return length * width;
}

int main()
{
    float length = 5.0;
    float width = 3.0;

```

```
15      // Function call
16      float area = calculateRectangleArea(length, width);
17
18      printf("The area of the rectangle is: %.2f\n", area);
19
20      return 0;
21  }
22
```

input Output

Success #stdin #stdout 0.01s 5436KB

```
The area of the rectangle is: 15.00
```

https://ideone.com/2AlaPi

Function Declaration

```c
// C program to illustrate the function declaration
#include <stdio.h>

// Function declartion
float calculateRectangleArea(float, float);

int main()
{
    float length = 20.0;
    float width = 3.0;

    // Function call
    float area = calculateRectangleArea(length, width);

    printf("The area of the rectangle is: %.2f\n", area);

    return 0;
}
```

```
19
20  // Function definition
21  float calculateRectangleArea(float length, float width)
22  {
23      return length * width;
24  }
25
```

https://ideone.com/TwJtef

stdout

```
The area of the rectangle is: 60.00
```

Function without return type and without arguments

```
1   #include<stdio.h>
2   void printName();  //prototype
3   void main ()
4   {
5       printf("Hello ");
6       printName();
7   }
8   void printName()
9   {
10      printf("Welcome to CSC-101 at IITR");
11  }
12
```

stdout

```
Hello Welcome to CSC-101 at IITR
```

https://ideone.com/m1jkcp

- Write a c program to calculate nc_r using functions.

```c
#include <stdio.h>

// Function to calculate the factorial of a number
unsigned long long factorial(int num) {
    unsigned long long fact = 1;

    for (int i = 1; i <= num; i++) {
        fact *= i;
    }

    return fact;
}

```

https://ideone.com/aS8KyR

```
14  // Function to calculate n choose r (nCr)
15  unsigned long long nCr(int n, int r) {
16      if (n < 0 || r < 0 || r > n) {
17          return 0; // Invalid input
18      }
19
20      unsigned long long numerator = factorial(n);
21      unsigned long long denominator = factorial(r) * factorial(n - r);
22
23      return numerator / denominator;
24  }
25
```

```
26  int main() {
27      int n, r;
28
29      printf("Enter the value of n: ");
30      scanf("%d", &n);
31
32      printf("Enter the value of r: ");
33      scanf("%d", &r);
34
35      unsigned long long result = nCr(n, r);
36
37      printf("%dC%d = %llu\n", n, r, result);
38
39      return 0;
40  }
```

stdin

```
10 3
```

stdout

```
Enter the value of n: Enter the value of r: 10C3 = 120
```

Maximum of two numbers

</> source code

```c
#include <stdio.h>

// Function to find the maximum of two numbers
int findMaximum(int num1, int num2) {
    if (num1 > num2) {
        return num1;
    } else {
        return num2;
    }
}

```

https://ideone.com/STvwQQ

```c
int main() {
    int num1, num2;

    // Input the two numbers
    printf("Enter the first number: ");
    scanf("%d", &num1);

    printf("Enter the second number: ");
    scanf("%d", &num2);

    // Call the function to find the maximum and print the result
    int max = findMaximum(num1, num2);

    printf("Maximum number is: %d\n", max);

    return 0;
}
```

Enter the first number: 20
Enter the second number: 40
Maximum number is: 40

Call by or Pass by Value

```c
#include <stdio.h>

// Function to swap two integers using call by value
void swapByValue(int num1, int num2) {
    int temp = num1;
    num2 = num1;
    num2 = temp;
}

int main() {
    int num1, num2;

    printf("Enter the first integer: ");
    scanf("%d", &num1);

```

```
16      printf("Enter the second integer: ");
17      scanf("%d", &num2);
18
19      printf("Before swapping: num1 = %d, num2 = %d\n", num1, num2);
20
21      // Call the swapByValue function to swap num1 and num2 (but it won't work)
22      swapByValue(num1, num2);
23
24      printf("After swapping (call by value): num1 = %d, num2 = %d\n", num1, num2);
25
26      return 0;
27  }
28
```

https://ideone.com/BAr1zQ

```
Enter the first integer: 100
Enter the second integer: 50
Before swapping: num1 = 100, num2 = 50
After swapping (call by value): num1 = 100,
num2 = 50
```

Thank You!