Presented By:Mostafa Saqly

C Functions

Content

- Types of function
- Types of User-defined Functions
- C Recursion
- C Storage Class

Types of function

- A function Definition : is a block of code that performs a specific task.
- There are two types of function in C programming:

1. Standard library functions
2. User-defined functions

Standard library functions

- The standard library functions are built-in functions in C programming.
- These functions are defined in header files. **For example :**
- The printf() is a standard library function to send formatted output to the screen (display output on the screen). This function is defined in the stdio.h header file.
  Hence, to use the printf()function, we need to include the stdio.h header file using #include <stdio.h>.
- The sqrt() function calculates the square root of a number. The function is defined in the math.h header file.

User-defined function

- You can also create functions as per your need. Such functions created by the user are known as user-defined functions.

How user-defined function works?

- #include <stdio.h>
- void functionName()
- { ... . }
- int main() { ... .
- functionName();
- ... . }

User-defined function

- The execution of a C program begins from the main() function.
- When the compiler encounters functionName();, control of the program jumps to
- void functionName()And, the compiler starts executing the codes inside functionName().
- The control of the program jumps back to the main() function once code inside the function definition is executed.

Advantages of user-defined function

- The program will be easier to understand, maintain and debug.
- Reusable codes that can be used in other programs
- A large program can be divided into smaller modules. Hence, a large project can be divided among many programmers.

C User-defined functions

- **Function prototype**
- A function prototype is simply the declaration of a function that specifies function's name, parameters and return type. It doesn't contain function body.
- A function prototype gives information to the compiler that the function may later be used in the program.

C User-defined functions

Syntax of function prototype

- returnType functionName(type1 argument1, type2 argument2, ...);In the above example, int addNumbers(int a, int b); is the function prototype which provides the following information to the compiler:
- name of the function is addNumbers()
- return type of the function is int
- two arguments of type int are passed to the function
- The function prototype is not needed if the user-defined function is defined before the main() function.

C User-defined functions

Calling a function

- Control of the program is transferred to the user-defined function by calling it.
- **Syntax of function call**
- functionName(argument1, argument2, ...);In the above example, the function call is made using addNumbers(n1, n2); statement inside the main() function.

Passing arguments to a function

How to pass arguments to a function?

```
#include <stdio.h>

int addNumbers(int a, int b);

int main()
{
    ... .. ...

    sum = addNumbers(n1, n2);

    ... .. ...
}

int addNumbers(int a, int b)
{
    ... .. ...
    ... .. ...
}
```

Return Statement

Return statement of a Function

```
#include <stdio.h>

int addNumbers(int a, int b);

int main()
{
    ... .. ...

    sum = addNumbers(n1, n2);

    ... .. ...
}

int addNumbers(int a, int b)
{
    ... .. ...
    return result;
}
```

sum = result

Types of User-defined Functions in C

- No arguments passed and no return value.
- No arguments passed but a return value
- Argument passed but no return value
- Argument passed and a return value

C Recursion

- A function that calls itself is known as a recursive function. And, this technique is known as recursion.
-

How recursion works?

```
void recurse()
{
    ... .. ...
    recurse();
    ... .. ...
}

int main()
{
    ... .. ...
    recurse();
    ... .. ...
}
```

recursive call

Advantages and Disadvantages of Recursion

- Recursion makes program elegant. However, if performance is vital, use loops instead as recursion is usually much slower.
- That being said, recursion is an important concept. It is frequently used in data structure and algorithms. For example, it is common to use recursion in problems such as tree traversal.

C Storage Class

- Every variable in C programming has two properties: type and storage class.
- Type refers to the data type of a variable. And, storage class determines the scope, visibility and lifetime of a variable.

There are 4 types of storage class:

- automatic
- external
- static
- register

C Storage Class

Local Variable :

- The variables declared inside a block are automatic or local variables. The local variables exist only inside the block in which it is declared.

Global Variable

- Variables that are declared outside of all functions are known as external or global variables. They are accessible from any function inside the program.

C Storage Class

Register Variable

- The register keyword is used to declare register variables. Register variables were supposed to be faster than local variables.
- However, modern compilers are very good at code optimization, and there is a rare chance that using register variables will make your program faster.
- Unless you are working on embedded systems where you know how to optimize code for the given application, there is no use of register variables.

C Storage Class

Static Variable

- A static variable is declared by using the static keyword. For example;
- static int i;
- The value of a static variable persists until the end of the program.

Thanks!

Any questions?