Chapter 6 - Modularity Using Functions: Part I

At a Glance

Lesson Contents

- Overview
- Objectives
- Quick Quizzes
- Additional Resources
- Key Terms
- Lesson Assignment

Chapter Notes

Overview

Chapter 6 introduces the role of C functions. You learn about functions and parameter declarations, and also how to return a value. In the case study, you put to practice the concepts learned to the particular case of calculating age norms. You also learn about several useful standard library functions. Finally, you learn to identify and avoid common programming and compiler errors.

Objectives

- Function and parameter declarations
- Returning a value
- Case study: Calculating age norms
- Standard library functions
- Common programming and compiler errors

Function and Parameter Declarations

Topic Tip	The Programming Note on page 279 provides a concise explanation of the difference between a function prototype, the calling statement and a function header line.

Topic Tip	It is important to note that in earlier versions of C, function prototypes were not required. Additionally, if a function header line omitted a return data type, the return value was, by default, implicitly declared as being of type `int`. For more information, see the Programming Note on page 280.

Topic Tip	Note the importance of providing preconditions and postconditions (see the Programming Note on page 284).

Quick Quiz 1

1. What is the difference between a called function and a calling function?
2. The items enclosed within the parentheses in a function call statement are called _____of the function.
3. What is a pass by value?
4. The argument names in the header line of a function are known as _____.

Returning a Value

Topic Tip	A basic rule of testing states that each function should only be tested in a program in which all other functions are known to be correct (see the Programming Note on page 295).

Quick Quiz 2

1. A(n)_____is the beginning of a final function that is used as a placeholder for the final function until the function is completed.
2. Pass by value is also referred to as_____.
3. How can you return a value from a function?
4. How do you write the prototype of a function with an empty parameter list?

Standard Library Functions

Quick Quiz 3

1. What are random numbers?

2. What are pseudorandom numbers?

3. The method for adjusting the random numbers produced by a random-number generator to reside within a specified range is called_____.

4. The standard library consists of_____header files.

Additional Resources

1. C Tutorial: Functions:
   https://www.w3schools.in/c-tutorial/functions/

2. C Programming Tutorial: Lesson 4: Functions:
   www.cprogramming.com/tutorial/c/lesson4.html

3. C Standard Library:
   http://en.wikipedia.org/wiki/C_standard_library

4. Using Pseudorandom Numbers:Guess Game using rand() and srand() in C
   https://www.geeksforgeeks.org/guess-game-using-rand-srand-c/

Key Terms

- Other terms used as synonyms for arguments are **actual arguments实际参数** and **actual parameters实参**.
- A fairly common procedure in child development is to establish normal ranges for height and weight as they relate to a child's age; these normal ranges are frequently referred to as **age norms正常年龄标准值**.
- The items enclosed within the parentheses in a function call statement are called **arguments实参** of the function.
- A function that is called or summoned into action by its reference in another function is a **called function被调用函数**.
- A function that calls another function is referred to as the **calling function调用函数**.
- The purpose of a **function body函数体** is to operate on the passed data and return, at most, one value directly back to the calling function.

- The portion of the function header that contains the function name and parameters is known as a **function declarator函数声明部**, which should not be confused with a function declaration (prototype).
- The purpose of a **function header函数头** is to identify the data type of the value returned by the function, if any, provide the function with a name, and specify the number, order, and type of values expected by the function.
- A **function prototype函数原型** declares the function to the compiler—it tells the compiler the name of the function, the data type of the value that the function will return (the keyword `void` indicates that the function will not be returning any value), and the data types of each argument that the function expects to receive when it is called.
- The argument names in the header line of a function are known as **parameters参数** or **formal parameters形式参数** and **formal arguments形参**.
- When a function simply receives copies of the values of each of the arguments and must determine where to store these values before it does anything else, this is known as a **pass by value传值** (or a **call by value传值调用**).
- **Pseudorandom numbers伪随机数** are numbers which are not really random, but are sufficiently random for the task at hand.
- **Random numbers随机数** are a series of numbers whose order cannot be predicted.
- The method for adjusting the random numbers produced by a random-number generator to reside within a specified range is called **scaling缩放比例**.
- A **stub桩子** is the beginning of a final function that is used as a placeholder for the final function until the function is completed.