
コース: C Programming

C Programming

作家: Anrich Tait

Abstract

Quick explanation of document.

Contents

1	Overview of C	2
1.1	Objectives:	2

Chapter 1

Overview of C

1.1 Objectives:

1. Become familiar with general form of a C program and it's basic elements,
2. Why you should write comments
3. Use of data types and the differences between the various data types.
4. How to declare variables
5. How to change the values of variables
6. Evaluate arithmetic expressions
7. Read data values into a program and display them
8. Understand strings
9. Redirection to use files for input/output
10. Understand the differences between runtime errors, syntax errors and logic errors, and how to debug each.

Unsorted Notes:

1. Both `#define` and `#include` are handled by the pre-processor, this is why we cannot change a variable or value that has been `#define`. The compiler is not capable of going back to change it.

2. Your variable names are also known as identifiers.

Steps in the compiling process:

1. preprocessing: Scans header files for relative prototypes. (So the compiler knows what printf is). Also looks for variables with #define
2. compiling: Turn code into assembly language before it is turned into 0's and 1's.
3. assembling: Where the assembly language is turned into machine code.
4. linking: Combines all the machine code into the final program that can be executed as your program.

Data Types:

1. int : short for integer. An int can be any whole number between -32767 and 32767.
2. double : Basically a real number, which has an integral part and a fractional part that is separated by a decimal point. For example: 3.14159; 0.0005; 150.05. Scientific notation can be used for doubles, for example: the real number

$$1.23 * 10^5$$

is equivalent to 123000.0 where the exponent '5' means "move the decimal point 5 places to the right. In Scientific notation this number is written as 1.23E5. Read the letter *e* or *E* as "times 10 to the power": 1.23e5 means 1.23 times 10 to the power of 5. If the exponent has a minus sign the decimal point is moved to the left.

Only use double when necessary, using the int data type is faster in most cases. Also int computations are always precise whereas double numbers can have rounding errors.