

Objectives

At the end of this session, you will be able to:

- Discuss variables
- Differentiate between variables and constants
- List the different data types and make use of them in C programs
- Discuss arithmetic operators

2.1 Variables

Modern day languages enable us to use symbolic names known as variables, (các ngôn ngữ hiện đại cho phép chúng ta sử dụng các tên ký hiệu được gọi là biến). to refer to the memory location where a particular value is to be stored. (để chỉ đến vị trí bộ nhớ nơi một giá trị cụ thể sẽ được lưu trữ)

2.2 Constants

Hằng số

A constant is a value whose worth never changes. (Một hằng số là một giá trị mà giá trị của nó không bao giờ thay đổi.)

2.3 Identifier

Mã định danh

The names of variables, functions, labels, and various other user-defined objects are called identifiers. (Tên của các biến, hàm, nhãn và nhiều đối tượng do người dùng định nghĩa khác được gọi là các định danh). These identifiers can contain one or more characters. (Các định danh này có thể chứa một hoặc nhiều ký tự). It is compulsory that the first character of the identifier is a letter or an underscore (_). (**Điều bắt buộc là ký tự đầu tiên của định danh phải là một chữ cái hoặc một dấu gạch dưới (_)). ** The subsequent characters can be alphabets, numbers, or underscores. (Các ký tự tiếp theo có thể là chữ cái, số hoặc dấu gạch dưới.)

2.3.1 Guidelines for Specifying Identifier Names

Hướng dẫn chỉ định tên định danh

However, some conventions that are typically followed are (Tuy nhiên, một số quy ước thường được tuân theo là):

- Variable names must begin with an alphabet. _Tên biến phải bắt đầu bằng một chữ cái.
- The first character may be followed by a sequence of letters or digits and can also include a special character like (Ký tự đầu tiên có thể được theo sau bởi một chuỗi các chữ cái hoặc số và cũng có thể bao gồm một ký tự đặc biệt như.)
 - Avoid using the letter O in places where it can be confused with the number 0, _Tránh sử dụng chữ O ở những chỗ có thể bị nhầm với số 0, and similarly, the lowercase letter L can be mistaken for the number 1. (và tương tự, chữ cái thường l có thể bị nhầm với số 1)
- Proper names should be avoided while naming variables. (Nên tránh sử dụng tên riêng khi đặt tên cho các biến.)

- Typically, uppercase and lowercase letters are treated as different, (*Thông thường, chữ hoa và chữ thường được coi là khác nhau.*), variables ADD, add, and Add are all different. *_ tức là, các biến ADD, add và Add đều là khác nhau.*

- As case-sensitivity considerations vary with programming languages, *Vì các quy tắc phân biệt chữ hoa chữ thường khác nhau giữa các ngôn ngữ lập trình*, it is advisable to maintain a standard way of naming variables. *nên tốt nhất là duy trì một cách đặt tên biến theo tiêu chuẩn*

- A variable name should be meaningful and descriptive; it should describe the kind of data it holds. *Tên biến nên có ý nghĩa và mô tả; nó nên mô tả loại dữ liệu mà nó chứa.* For example, if the sum of two numbers is to be found, *Ví dụ, nếu tổng của hai số cần được tìm*, the variable storing the result may be called sum. *biến lưu trữ kết quả có thể được gọi là sum.* Naming it s or ab12 is not a good idea. *Việc đặt tên là s hoặc ab12 không phải là một ý hay*

2.3.2 Keywords

Từ khóa

we need to ensure that we do not use one of these keywords as a variable name (*chúng ta cần đảm bảo rằng không sử dụng một trong những từ khóa này làm tên biến.*)

Some programming languages require the programmer to specify the name of the variable as well as the type of data that is to be stored in it, before actually using a variable (*Một số ngôn ngữ lập trình yêu cầu lập trình viên phải chỉ định tên biến cũng như loại dữ liệu sẽ được lưu trữ trong đó trước khi thực sự sử dụng biến.*). This step is referred to as 'variable declaration' (*Bước này được gọi là 'khai báo biến'.*).

2.4 Data types

Kiểu dữ liệu

The general form of declaring a variable is (*Dạng chung để khai báo một biến là:*):

```
Data type (variable name)
```

Notes

Kiểu dữ liệu

- int (số nguyên) : phạm vi -32768 đến 32767
hiển thị = %d 2 byte = 16 bit
- float (số thực) : gồm số nguyên và thập phân
hiển thị = %f 4 byte = 32 bit
- double (số thực rộng = 2 float)

hiển thị = %lf 8 byte = 64 bit

- char (1 ký tự) : lưu trữ 1 ký tự
bao quanh bởi ngoặc đơn ' '

hiển thị = %c

- short int = int = %d

long int = %ld

Khai báo chuỗi dữ liệu

char **tenbuen** [] = "....."

Hiển thị

printf (" %s", **tenbuen**)



2.4.1 Basic and Derived Data types

Kiểu dữ liệu cơ bản và phái sinh

Modifiers used with C are signed, unsigned, long and short (*Các trình sửa đổi được sử dụng với C là signed, unsigned, long và ngắn*)

The signed and unsigned Types (*Các loại có dấu và không dấu*)

Default integer declaration assumes a signed number. (*Khai báo số nguyên mặc định giả định một số có dấu.*) The most important use of signed is to modify the char data type, (*Sự sử dụng quan trọng nhất của signed là để sửa đổi kiểu dữ liệu char*), where char is unsigned by default. (*trong đó char mặc định là không có dấu.*)

The unsigned type specifies that a variable can take only positive values. Kiểu không có dấu xác định rằng một biến chỉ có thể nhận các giá trị dương.

By prefixing the int type with the word unsigned, the range of positive numbers can be doubled (*Bằng cách thêm tiền tố unsigned vào kiểu int, phạm vi số dương có thể được nhân đôi.*)

The long and short Types (*Các loại dài và ngắn*)

A long int variable is declared as follows (*Biến int dài được khai báo như sau*):

```
long int varNum;
```

It can also be declared simply as long varNum as well (*Một số nguyên dài có thể được khai báo là long int hoặc chỉ là long*). A long integer can be declared as long int or just long (*Nó cũng có thể được khai báo đơn giản là long varNum.*). Similarly, a short integer can be declared as short int or short (*Tương tự, một số nguyên ngắn có thể được khai báo là short int hoặc short.*).

Notes

Mixed Mode Expression

(Biểu thức hỗn hợp)

Thứ tự các kiểu dữ liệu

$\text{char} < \text{int} < \text{long} < \text{float} < \text{double}$

Type Conversions

(Chuyển dữ liệu)

+ char, short \rightarrow int

+ float \rightarrow double

+ double + double \rightarrow double

+ long + long \rightarrow double

+ unsigned + unsigned \rightarrow unsigned

+ long + unsigned \rightarrow unsigned long

Cast (ép kiểu)

(type) cast



Notes

Ex01: khai báo biến : Kiểu dữ liệu (tên biến)

Ex02: dãy số

Ex03: nhập 1 ký tự thường, hiển thị
ký tự hoa

- + gán biến char ch;

- + Hiển thị printf("...", ch (ch - 32));

Ex04: nhập 1 ký tự thường, hiển thị
ký tự hoa bằng hàm toupper trong
thư viện ctype.h

```
#include <ctype.h>
```

```
printf("...", toupper(ch));
```

→ chuyển từ hoa sang thường = hàm
tolower()





Summary

- Most often, an application needs to handle data; it needs some place where this data can be temporarily stored. This 'place' where this data is stored is called the memory.
- Modern day languages enable us to use symbolic names known as variables, to refer to the memory location where a particular value is to be stored.
- There is no limit to the number of memory locations that a program can use.
- A constant is a value whose worth never changes.
- The names of variables, functions, labels, and various other user-defined objects are called identifiers.
- All languages reserve certain words for their internal use. They are called keywords.
- The main data types of C are character, integer, float, double float and void.
- Modifiers are used to alter the basic data types so as to fit into various situations. Unsigned, short and long are the three modifiers available in C.
- C supports two types of Arithmetic operators: Unary and Binary.
- Increment(++) and decrement(--) are unary operators acting only on numeric variables.
- Arithmetic binary operations are + - * / % which act only on numeric constants, variables or expressions.
- The modulus operator % acts only on integers and results in remainder after integer division.