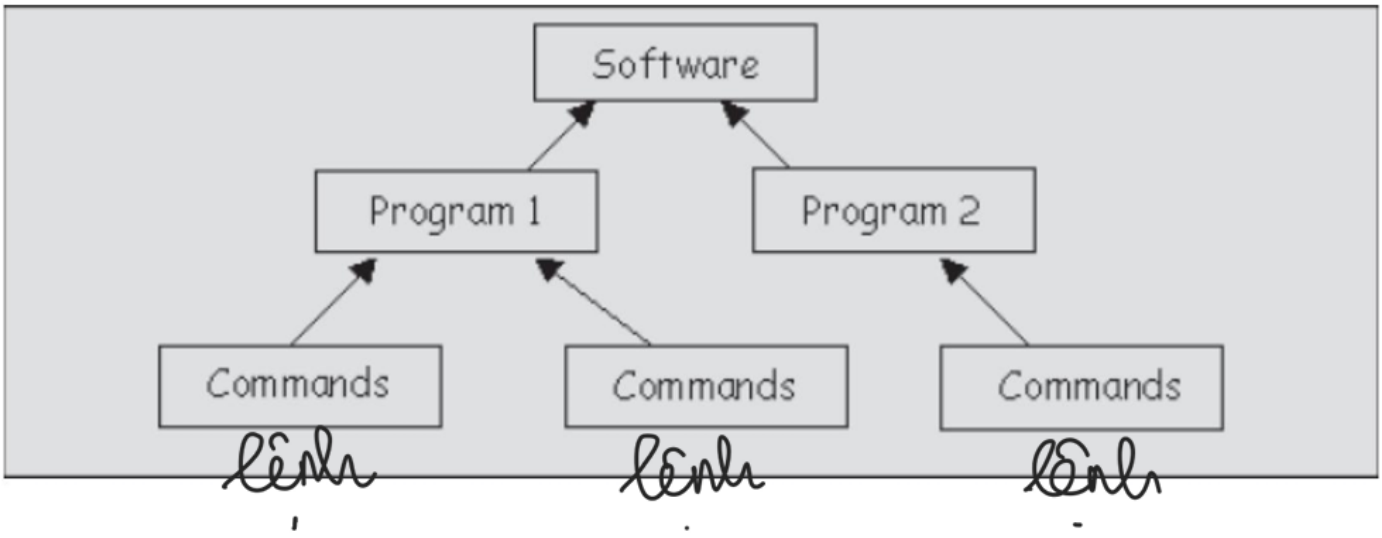
Objectives

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

- *Differentiate(phân biệt) between Command, Program and Software*
- *Explain(giải thích) the beginning of C*
- *Explain when and why is C used*
- *Discuss(thảo luận) the C program structure(cấu trúc chương trình)*
- *Discuss algorithms(thuật toán)*
- *Draw flowcharts(sơ đồ luồng công việc)*
- *List the symbo(kí hiệu)ls used in flowcharts*

1.1 Instructions to a Computer



Hình 1.1: Phần mềm, Chương trình và Lệnh

1.2 The C Language

C was used for systems programming(*C được sử dụng để lập trình hệ thống)*

Operating Systems(*hệ điều hành*), Interpreters(*trình thông dịch*), Editors(*trình biên tập*), Assembly programs(*chương trình lắp ráp*) are usually called systems programs(*thường được gọi là chương trình hệ thống)*

UNIX Operating System was developed using C(*Hệ điều hành UNIX được phát triển bằng C*)

C compiler produces fast and error-free object code. (*Trình biên dịch C tạo ra mã đối tượng nhanh và không có lỗi)*

C code is very portable(*mã C rất dễ di chuyển*),

1.2.2 C - A Structured Language

Ngôn ngữ có cấu trúc

C allows synthesis of code and data (*C cho phép tổng hợp mã và dữ liệu*.).

Functions (*hàm)* are used to define and separate, tasks required in a program (*được sử dụng để xác định và tách biệt các nhiệm vụ cần thiết trong một chương trình).*

Code block (*Khối mã*) is a logically connected group of program statements that is treated like a unit (*là một nhóm các câu lệnh chương trình được kết nối logic được xử lý như một đơn vị)*. A code block is created by placing a sequence of statements between opening and closing curly braces as shown below (*Một khối mã được tạo ra bằng cách đặt một chuỗi các câu lệnh giữa dấu ngoặc nhọn mở và đóng như minh họa bên dưới)*

```
do
{
  i = i + 1;
  .
  .
  .
} while (i < 40);
```

1.3 The C Program Structure

Cấu trúc chương trình

C has few keywords, 32 to be precise (*C có ít từ khóa, chính xác là 32).*

Some rules(qui tắc) for programs written in C are as follows:

- All keywords are lower cased (Tất cả các từ khóa đều được viết thường)
- C is case sensitive, do while is different from DO WHILE (C phân biệt chữ hoa chữ thường, do while khác với DO WHILE)

- Keywords cannot be used for any other purpose, that is, they cannot be used as a variable or function name (Từ khóa không thể được sử dụng cho bất kỳ mục đích nào khác, nghĩa là chúng không thể được sử dụng như một biến hoặc tên hàm)
- main() is always the first function called when a program execution begins (main luôn là hàm đầu tiên được gọi khi chương trình bắt đầu thực thi

1.3.1 Function Definition

Định nghĩa hàm

The function name is always followed by parentheses .(*Tên hàm luôn được theo sau bởi dấu ngoặc đơn).* The parentheses may or may not contain parameters (*Dấu ngoặc đơn có thể chứa hoặc không chứa tham số).*

1.3.2 Delimiters

Dấu phân cách

The function definition is followed by an pen curly brace ({) *(Định nghĩa hàm được theo sau bởi dấu ngoặc nhọn mở ({)).*This curly brace signalsthe beginning of the function (*Dấu ngoặc nhọn này báo hiệu sự bắt đầu của hàm*). Similarly a closing curly brace (}) after the statements, in the function, indicate the end of the function (một dấu ngoặc nhọn đóng (}) sau các câu lệnh trong hàm, chỉ ra sự kết thúc của hàm)

1.3.3 Statement Terminator

Bộ kết thúc câu lệnh

A statement in C is terminated with a semicolon (;) *(Một câu lệnh trong C được kết thúc bằng dấu chấm phẩy (;))*A carriage return, whitespace, or a tab is not understood by the C compiler (*ký tự trả về, khoảng trắng hoặc tab : trình biên dịch C không hiểu được.*

There can be more than one statement on the same line as long as each one of them is terminated with a semi-colon (*Có thể có nhiều hơn một câu lệnh trên cùng một dòng miễn là mỗi câu lệnh được kết thúc bằng dấu chấm phẩy.)* . A statement that does not end in a semicolon is treated as an invalid line of code in C (*Một câu lệnh không kết thúc bằng dấu chấm phẩy được coi là một dòng mã không hợp lệ trong C.)*

1.3.4 Comment Lines

Dòng chú thích

The compiler ignores them. In C, comments begin with /* and are terminated with */(*Trong C, chú thích bắt đầu bằng /* và kết thúc bằng */),* in case the comments contain multiple lines(t*rong trường hợp chú thích chứa nhiều dòng.).* Care should be taken that the terminating delimiter (*/) is not forgotten(*Cần lưu ý không quên dấu phân cách kết thúc (*/)).* Otherwise, the entire program will be treated like a comment(*Nếu không, toàn bộ chương trình sẽ được coi như một bình luận).* In case the comment contains just a single line you can use // to indicate that it is a comment*(Trong trường hợp bình luận chỉ chứa một dòng duy nhất, bạn có thể sử dụng // để chỉ ra rằng đó là một bình luận.).*

For example

```
int a=0; //Variable 'a' has been declared as an integer data type
```

1.4 Compiling and Running a Program

Biên dịch và chạy chương trình

The various stages of translation of a C program from source code to executable code are as follows: *Các giai đoạn khác nhau trong quá trình dịch một chương trình C từ mã nguồn sang mã thực thi như sau:*

¬ Editor/Word Processor (*Biên tập viên/Bộ xử lý văn bản)*

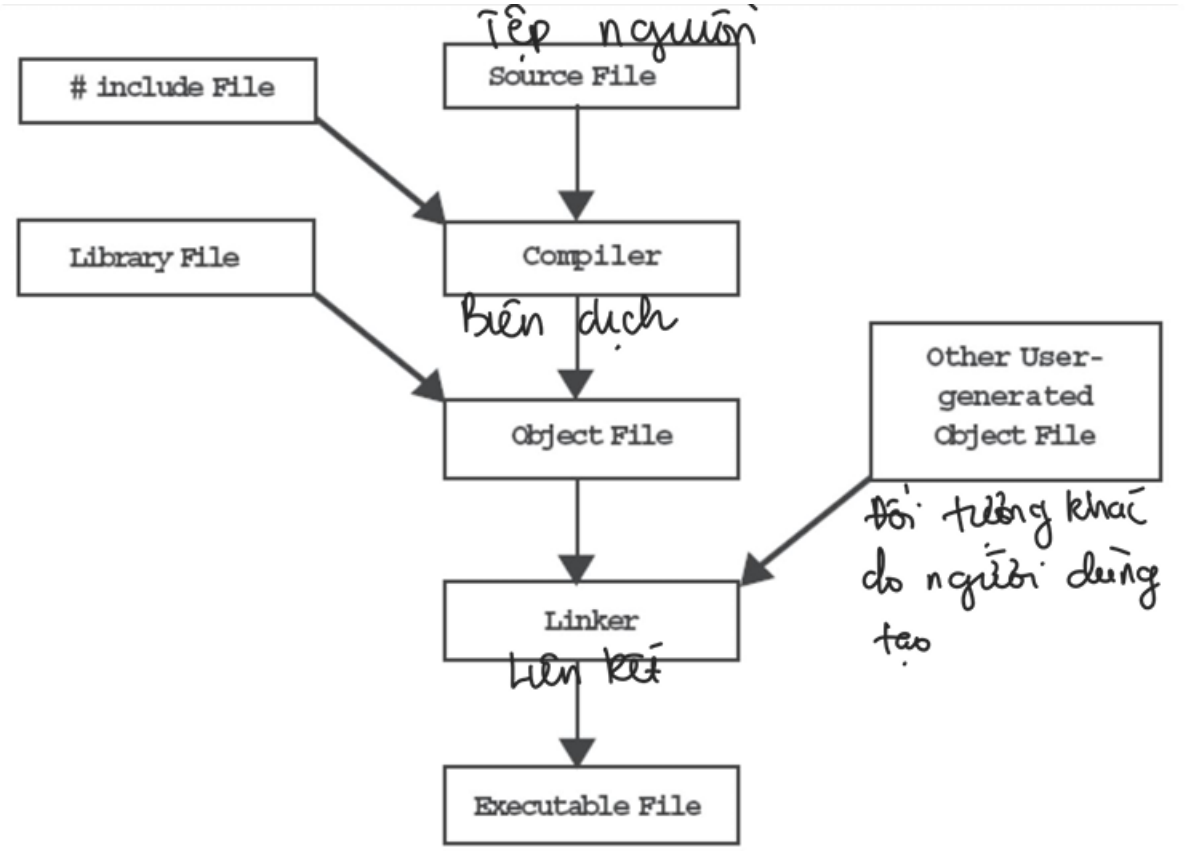
The source code is written using an editor or a word processor*.(Mã nguồn được viết bằng một trình soạn thảo hoặc chương trình xử lý văn bản.).* The code should be written in the form of standard text files, (*Mã phải được viết dưới dạng tệp văn bản tiêu chuẩn)*.Some compilers supply programming environments that include an editor (*Một số trình biên dịch cung cấp môi trường lập trình bao gồm cả trình soạn thảo văn bản)*

¬ Source Code (*mã nguồn*)

This is the text of the program, which the user can read.(*Đây là văn bản của chương trình, mà người dùng có thể đọc được*.)It is the input for the C compiler.(*Đây là đầu vào cho trình biên dịch C)*

¬ C Preprocessor (*Bộ tiền xử lý)*

The source code is first passed through the C preprocessor.(*Mã nguồn đầu tiên được chuyển qua bộ tiền xử lý C.*)Preprocessors act on statements beginning with #.(*Các bộ tiền xử lý hoạt động trên các câu lệnh bắt đầu bằng #).*These statements are called directives (explained later).(*Những câu lệnh này được gọi là chỉ thị)*. The directives are usually placed at the start of the program, though they can be placed anywhere else.(*Các chỉ thị thường được đặt ở đầu chương trình, mặc dù chúng có thể được đặt ở bất kỳ đâu).*The directives are short names given to a set of code.(*Các chỉ thị là những tên ngắn được gán cho một tập hợp mã lệnh)*



Hình 1.2: Biên dịch và chạy chương trình

1.5.1 Pseudo code

Mã giả

.Note that ‘pseudo code’ is not actual code (pseudo=false) (*Lưu ý rằng 'mã giả' không phải là mã thực sự (pseudo = sai)).*Pseudo code uses a certain standard set of words, which makes it resemble a code (*Mã giả sử dụng một tập hợp từ chuẩn nhất định, khiến nó trông giống như mã lệnh*).However, unlike code, pseudo code cannot be compiled or run (*Tuy nhiên, không giống như mã lệnh, mã giả không thể được biên dịch hoặc chạy).*

Example 2:

```
BEGIN
    INPUT A, B
    DISPLAY A + B
END
```

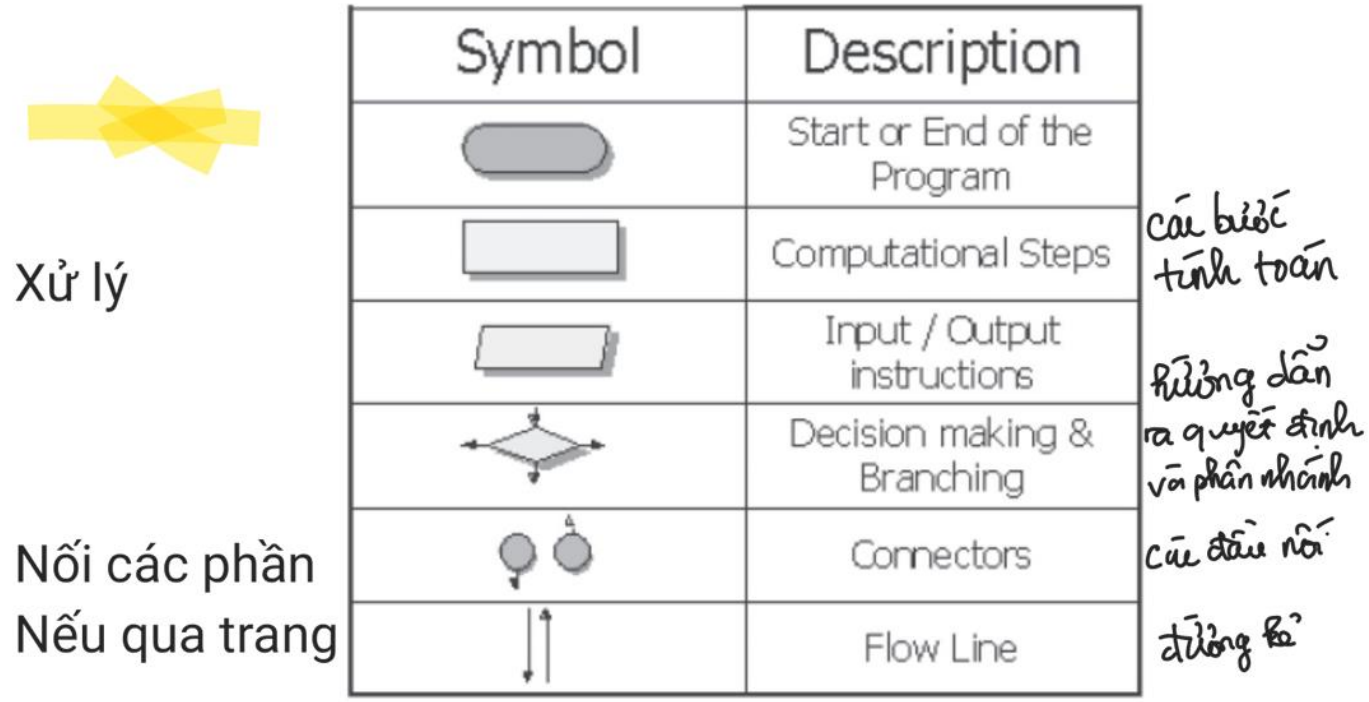
In this pseudo code, the user inputs two values, which are stored in memory and can be accessed as A and B respectively *(Trong mã giả này, người dùng nhập hai giá trị, được lưu trữ trong bộ nhớ và có thể truy cập tương ứng là A và).* Such named locations in memory are called variables *(Các vị trí được đặt tên như vậy trong bộ nhớ được gọi là biến).*

A set of instructions or steps in a pseudo code is collectively called a construct (*Một tập hợp các hướng dẫn hoặc các bước trong mã giả được gọi chung là một cấu trúc*). There are three types of

programming constructs - sequence, selection, and iteration constructs (*Có ba loại cấu trúc lập trình - cấu trúc tuần tự, cấu trúc lựa chọn và cấu trúc lặp)*

1.5.2 Flowcharts

Sơ đồ luồng



Hình 1.4: Biểu tượng sơ đồ luồng

Some other essential things to be taken care of when drawing are:*(Một số điều cần thiết khác cần lưu ý khi vẽ sơ đồ luồng công việc là):*

- Initially concentrate a only on the logic of the problem and drawout the main path of the f lowchart (*Ban đầu chỉ tập trung vào logic của vấn đề và vạch ra lộ trình chính của sơ đồ.)*
- A flowchart must have only one STARTand one STOP point (*Sơ đồ luồng công việc chỉ được có một điểm BẮT ĐẦU và một điểm DỪNG.)*
- It is not necessary to represent each and every step of aprogram in the flowchart. Only the essential and meaningful steps need to be represented (*Không cần thiết phải thể hiện từng bước của một chương trình trong sơ đồ luồng. Chỉ có các bước thiết yếu và có ý nghĩa cần phải được trình bày)*

Summary

Software is a set of programs.Phần mềm là một tập hợp các chương trình.
A program is a set of instructions.Một chương trình là một tập hợp các hướng dẫn.
Code blocks form the base of any C program.Các khối mã tạo thành nền tảng của bất kỳ chương trình C nào.

The C language has 32 keywords.Ngôn ngữ C có 32 từ khóa.
Steps involved in solving a problem are studying the problem in detail, gathering the relevant information, processing the information, and arriving at the results.Các bước liên quan đến việc giải quyết một vấn đề bao gồm nghiên cứu chi tiết vấn đề, thu thập thông tin liên quan, xử lý thông tin và đi đến kết quả.

An algorithm is a logical and concise list of steps to solve a problem.Thuật toán là một danh sách các bước hợp lý và ngắn gọn để giải quyết một vấn đề.
Algorithms are written using pseudo codes or flowcharts.Các thuật toán được viết bằng mã giả hoặc biểu đồ dòng chảy.

A pseudo code is a representation of an algorithm in a language that resembles code.Mã giả là một biểu diễn của một thuật toán trong một ngôn ngữ giống như mã.
A flowchart is a diagrammatic representation of an algorithm.Biểu đồ dòng chảy là một biểu diễn hình ảnh của một thuật toán.

The basic selection construct is an 'IF' construct.Cấu trúc chọn cơ bản là một cấu trúc 'IF'.
Flowcharts can be broken into parts, and connectors can be used to indicate the location of the joins.Biểu đồ dòng chảy có thể được chia thành các phần, và các đầu nối có thể được sử dụng để chỉ ra vị trí của các điểm nối.

When we come across a condition based on which the path of execution may branch, such constructs are referred to as selection, conditional, or branching constructs.Khi chúng ta gặp một điều kiện mà theo đó đường đi của việc thực hiện có thể phân nhánh, các cấu trúc như vậy được gọi là cấu trúc lựa chọn, điều kiện hoặc phân nhánh.

The IF…ELSE construct enables the programmer to make a single comparison and then execute the steps depending on whether the result of the comparison is True or False.Cấu trúc IF…ELSE cho phép lập trình viên thực hiện một phép so sánh đơn lẻ và sau đó thực hiện các bước tùy thuộc vào việc kết quả của phép so sánh là Đúng hay Sai.

A nested IF is an IF inside another IF statement.Một IF lồng nhau là một IF nằm trong một câu lệnh IF khác.

Often, it is necessary to repeat certain steps a specific number of times or till some specified condition is met.Thường thì cần lặp lại một số bước nhất định một số lần cụ thể hoặc cho đến khi một điều kiện nhất định được đáp ứng.The constructs which achieve these are known as iterative or looping constructs.Các cấu trúc đạt được điều này được gọi là các cấu trúc lặp hoặc vòng lặp.

Ex01: tính tổng A+B=C

Ex02: tính tổng A+B=C vs A, B nhập từ user

assigament operator: toán tử gán

trái (ghi) = phải (đọc)

Hàm getch() - 1 hàm trong thư viện conio.h

- dùng để đọc 1 ký tự từ bàn phím mà ko cần sd phím Enter
- nó hữu ích khi cần nhận 1 ký tự đầu vào và thực hiện 1 hành động ngay lập tức

Đặc điểm

- getch() ko hiển thị ký tự mà nhập trên màn hình
- Ko yêu cầu nhấn Enter sau khi nhập
- Dùng trong các ứng dụng điều khiển = phím chẳng hạn trò chơi console

Ex03: Số giai thừa - factorial number

$$n! = n(n-1)(n-2)(n-3)\ldots\ldots 1$$

Ex04: fibonacci number - là 1 dãy số trong toán học bắt đầu = 0 và 1, mỗi số tiếp theo

bằng cộng 2 số liền kề trước nó

VD: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

Công thức tổng quát

$F = F(n-1) + F(n-2)$

Operator $sum = n_1 + n_2$

sau đó gán $n_1 = n_2$

$n_2 = sum$

| rồi lặp lại

Ex05: amstrong number - là 1 số tự nhiên mà tổng các lũy thừa các chữ số của nó = chính nó

lũy thừa đc tính theo số lượng chữ số

VD: $153 = 1^3 + 5^3 + 3^3$

$9474 = 9^4 + 4^4 + 7^4 + 4^4$

Operator : VD: 9474

+ Tách các đơn vị chữ số:

• Hàng đơn vị 9474 % 10 (phép chia lấy dư)

9474 | 10 → dư 4

④ | 947 → gán biến mới

• Hàng chục 947 % 10 dư 7

947 | 10

⑦ | 94 → gán biến mới

. Hàng trăm 94 % 10 dư 4

94 | 10

(4) (9) → gán biến mới

. Hàng nghìn 9 % 10 dư 9

9 | 10

(9) 0

→ lấy ra đc 9 - 4 - 7 - 4

+ Đếm số chữ số : 4

+ lũy thừa các đơn vị chữ số $9^4 - 4^4 - 7^4 - 4^4$

Tính tổng sum = $9^4 + 4^4 + 7^4 + 4^4$

. So sánh sum = số đã nhập

⇒ Đây là số amstrong

Lab home - nhập 1 số và tính bình phương

Cách 1: pown = num * num

Cách 2: Dùng hàm lũy thừa pown trong thư viện math.h

```
# include <math.h>
    double result = pow (2, 9)
```

pow (số mũ, cơ số)

lab Rome _ tính chu vi _ diện tích hình tròn

$C = 2\pi r$

$S = \pi r^2$

Khai báo hằng số pi

define pi 3.14