ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

TIN HỌC ĐẠI CƯƠNG

Phần II: LẬP TRÌNH C

Nội dung chính

- Chương 1: Tổng quan về ngôn ngữ C
- Chương 2: Kiểu dữ liệu và biểu thức trong C
- Chương 3: Vào ra dữ liệu
- Chương 4: Cấu trúc điều khiển
- Chương 5: Mảng, con trỏ, xâu ký tự
- Chương 6: Cấu trúc
- Chương 7: Hàm
- Chương 8: Tệp dữ liệu

Chương 1: Tổng quan về ngôn ngữ C

- 1.1. Lịch sử phát triển của ngôn ngữ C
- 1.2. Các phần tử cơ bản của ngôn ngữ C
- 1.3. Cấu trúc cơ bản của chương trình C
- 1.4. Biên dịch chương trình C

Sự ra đời của C

- Nhu cầu viết lại HĐH Unix cho các hệ máy tính khác nhau
  - Dùng ngôn ngữ Assembly
    - Công việc nặng nề, phức tạp
    - Khó chuyển đổi chương trình giữa các hệ máy tính khác nhau
  - Cần ngôn ngữ mới
    - Đơn giản việc lập trình
    - Tính khả chuyển cao
- C ra đời tại Bell Lab thuộc tập đoàn AT&T
  - Tác giả Brian W. Kernighan & Dennis Ritchie
  - Dựa trên ngôn ngữ BCPL& B
  - Phát triển năm 1970, hoàn thành 1972

Ngôn ngữ lập trình C

- Đặc điểm
  - Ngôn ngữ lập trình hệ thống
  - Tính khả chuyển, linh hoạt cao
  - Có thế mạnh trong xử lý dữ liệu số, văn bản, cơ sở dữ liệu,..
- Phạm vi sử dụng
  - Viết các chương trình hệ thống
    - Hệ điều hành Unix có 90% mã C, 10% mã hợp ngữ
  - Các trình điều khiển thiết bị (device driver)
  - Xử lý ảnh

Ngôn ngữ lập trình C

- Các phiên bản
  - ANSI C: C chuẩn (1989)
  - Các phiên bản khác xây dựng dựa trên ANSI C
    - Đưa thêm thư viện; Bổ sung cho thư viện chuẩn của ANSI C
- Các trình biên dịch phổ biên
  - Microsoft Visual C/C++ (Microsoft)
  - gcc (GNU Project)

Chương 1: Tổng quan về ngôn ngữ C

1.1. Lịch sử phát triển của ngôn ngữ C

1.2. Các phần tử cơ bản của ngôn ngữ C

1.3. Cấu trúc cơ bản của chương trình C

1.4. Biên dịch chương trình C

Các phần tử cơ bản

- Tập ký tự
- Từ khóa
- Định danh
- Các kiểu dữ liệu
- Hằng
- Biến
- Hàm
- Biểu thức
- Câu lệnh
- Chú thích

1. Tập ký tự

- Ký tự là các phần tử cơ bản tạo nên chương trình
- Chương trình: Tập các câu lệnh nhằm giải quyết nhiệm vụ đặt ra
- Câu lệnh: là các từ (từ vựng) liên kết với nhau theo cú pháp của ngôn ngữ lập trình
  - Ví dụ: while (i < N) do
- Các từ: Tổ hợp các ký tự theo nguyên tắc xây dựng từ vựng
  - Ví dụ: TenFile, BaiTap2…

Tập ký tự trong C

- 26 chữ cái hoa: A B C ... X Y Z
- 26 chữ cái thường: a b c … x y z.
- 10 chữ số: 0 1 2 3 4 5 6 7 8 9.
- Các kí hiệu toán học: + - * / = < >
- Các dấu ngăn cách: . ; , : space tab
- Các dấu ngoặc: ( ) [ ] { }
- Các kí hiệu đặc biệt: _ ? $ & # ^ \ ! ‘ “ ~ ...

2. Từ khóa (keyword)

- Được định nghĩa sẵn trong C
- Dành riêng cho các mục đích xác định
  - Đặt tên cho kiểu dữ liệu:
    - int, float, double…
  - Mô tả các lệnh, các cấu trúc lập trình
    - if, else, while, case, for…

Từ khóa thông dụng

- Lưu ý: Tất cả từ khóa trong C đều viết bằng chữ cái thường

break	case	char	const	continue	default
do	double	else	enum	float	for
goto	if	int	interrupt	long	return
short	signed	sizeof	static	struct	switch
typedef	union	unsigned	void	while	

3. Định danh (Identifier)

- Định danh (Tên) là một dãy các kí tự dùng để gọi tên các đối tượng trong chương trình.
  - Các đối tượng trong chương trình
    - Biến
    - Hằng số
    - Hàm
    - Kiểu dữ liệu
- Định danh có thể được đặt bởi
  - Ngôn ngữ lập trình → các từ khóa
  - Người lập trình

Quy tắc đặt định danh trong C

- Định danh được bắt đầu bởi chữ cái hoặc dấu gạch dưới "_" (underscore)
- Các kí tự tiếp theo chỉ có thể là: chữ cái, chữ số hoặc dấu gạch dưới "_"
- Định danh do người lập trình đặt không được trùng với các từ khóa của C
- Độ dài định danh tùy thuộc phiên bản C
- Chú ý: C là ngôn ngữ có phân biệt chữ hoa và chữ thường

Ví dụ

- Định danh hợp lệ:
  - i, x, y, a, b, _function,
  - _MY_CONSTANT, PI, gia_tri_1
- Định danh không hợp lệ
  - 1_a, 3d, 55x (bắt đầu bằng chữ số)
  - so luong, sin() (có kí tự không hợp lệ, dấu cách, dấu ngoặc..)
  - int, char (trùng với từ khóa của C)

Một số quy ước (code convention)

- Định danh nên có tính gợi nhớ
- Nên sử dụng dấu gạch dưới để phân tách các định danh gồm nhiều từ
  - Có thể dùng cách viết hoa chữ cái đầu mỗi từ
    - Ví dụ: sinh_vien, sinhVien, SinhVien
- Quy ước thường được sử dụng:
  - Hằng số dùng chữ cái hoa
    - Ví dụ: PI, EPSILON,…
  - Các biến, hàm, cấu trúc dùng chữ cái thường
    - Biến điều khiển vòng lặp: i, j, k…
    - Hàm: NhapDuLieu, TimKiem,…
    - Cấu trúc: SinhVien, MatHang,…

4. Các kiểu dữ liệu

- Một kiểu dữ liệu là một tập hợp các giá trị mà một dữ liệu thuộc kiểu dữ liệu đó có thể nhận được.
  - Ví dụ: Một đối tượng kiểu *char* của C sẽ là
    - Một số nguyên (Số nguyên có dấu, 1 byte)
    - Giá trị thuộc khoảng: [-128…127]
- Trên một kiểu dữ liệu, xác định một số phép toán đối với các dữ liệu thuộc kiểu dữ liệu tương ứng.

Ví dụ kiểu int

Một số phép toán được định nghĩa trên kiểu dữ liệu số nguyên của C

Tên phép toán	Ký hiệu	Ví dụ
Đảo dấu	-	
Cộng; Trừ; Nhân	+ - *	
Chia lấy nguyên	/	17/3 → 5
Chia lấy phần dư	%	17%3 → 2
So sánh	> < >= <= == !=	
Phép toán trên bit: AND; OR; XOR; NOT, Shift,…	& | ^ ~ << >>	3^17 →18 ~3 → -4

5. Hằng

- Hằng là đại lượng có giá trị không đổi trong chương trình.
- Giá trị hằng do người lập trình xác định.
- Các khái niệm
  - Literals
  - Constant
  - Definition (macro)
- Các loại hằng
  - Hằng số nguyên
  - Hằng số thực
  - Hằng ký tự
  - Hằng chuỗi/xâu ký tự

Hằng số nguyên (int literal)

Trong C, hằng số nguyên có thể biểu diễn dưới các dạng: cơ số 10, 8, 16

Hệ 10	Hệ 16	Hệ 8
2011	**0x**7DB	**0**3733
396	**0x**18C	**0**614

Hằng số thực (floating point literal)

- Trong C, hằng số thực có thể biểu diễn dưới các dạng
  - Dạng số thực dấu phẩy tĩnh
  - Dạng số thực dấu phẩy động

Số thực dấu phẩy tĩnh	Số thực dấu phẩy động
3.14159	31.4159E-1
123.456	12.3456E+1 hoặc 1.23456E+2

Hằng ký tự (character literal)

- Hằng ký tự có thể biểu diễn theo hai cách
  - Đặt ký hiệu của ký tự giữa hai dấu nháy đơn
  - Dùng mã ASCII của ký tự:
    - Số thứ tự của ký tự đó trong bảng mã ASCII
    - Là số nguyên → tuân thủ quy tắc biểu diễn số nguyên

Ký tự	Dùng nháy đơn	Dùng mã ASCII
Chữ cái A	'A'	65, 0x41, 0101
Dấu nháy đơn	'\''	39, 0x27, 047
Ký tự tab	'\t'	9, 0x09, 011

Hằng xâu ký tự (string literal)

- Hằng xâu kí tự được biểu diễn bằng dãy các kí tự trong cặp dấu nháy kép.
- Ví dụ:
  - “ngon ngu lap trinh C”
  - “Tin hoc dai cuong”
  - “Dai hoc Bach Khoa Ha Noi”

6. Biến (variable)

- Biến là đối tượng lưu giữ dữ liệu và có thể thay đổi giá trị khi chương trình được thực thi.
- Biến được đặt (allocate) trong các ô nhớ thuộc bộ nhớ chính của máy tính.
- Số ô nhớ dành cho 1 biến phụ thuộc kiểu dữ liệu của biến.
  - VD mỗi biến kiểu float chiếm 4 byte trong bộ nhớ.
- Tên biến phải được đặt theo quy tắc đặt tên.

7. Hàm (function)

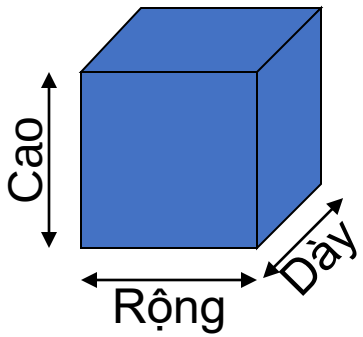
- Hàm là chương trình con có chức năng
  - Nhận dữ liệu đầu vào (các tham số vào)
  - Thực hiện một công việc nào đó
  - Có thể trả về một kết quả gọi là giá trị của hàm
    - Ví dụ: hàm sin(x)
      - sin(3.14/2) → 1.000
      - sin(3.14/6) → 0.499770
- Có 2 loại hàm:
  - ❖ Hàm có sẵn của các thư viện C cung cấp
  - ❖ Hàm do người lập trình tự định nghĩa
- Chi tiết về hàm sẽ được học ở phần sau

7. Hàm → Một số hàm toán học

Hàm	Ý nghĩa	Ví dụ
sqrt(x)	Căn bậc 2 của x	sqrt(16.0) → 4.0
pow(x,y)	x mũ y (x^y)	pow(2,3) → 8
fabs(x)	Trị tuyệt đối của x (|x|)	fabs(-5.0) → 5.0
exp(x)	E mũ x (e^x)	exp(1.0) → 2.71828
log(x)	Logarithm tự nhiên của x (*ln x*)	log(2.718) → 0.999
log10(x)	Logarithm cơ số 10 của x (*logx*)	log10(100) → 2.00
sin(x) cos(x)/ tan(x)	Các hàm lượng giác	
ceil(x)	Số nguyên nhỏ nhất không nhỏ hơn x ($\lceil x \rceil$)	ceil(2.5)=3 ceil(-2.5)=-2
floor(x)	Số nguyên lớn nhất không lớn hơn x ($\lfloor x \rfloor$)	floor(2.5)=2 floor(-2.5)=-3

8. Biểu thức

- Biểu thức là sự kết hợp các các toán hạng (operand) với các toán tử (operator) theo một quy tắc xác định.
  - Toán hạng có thể là biến, hằng, hàm, biểu thức…
  - Các toán tử rất đa dạng: cộng, trừ, nhân, chia…
- Khi biểu thức được tính toán sẽ trả về một kết quả.
- Ví dụ
  - Thể tích hình hộp: V = Rộng*Cao*Dày
    - Phép nhân (*) là toán tử
    - 3 toán hạng Rộng, Cao, Dày



9. Câu lệnh (statement)

- Câu lệnh diễn tả một hoặc một nhóm các thao tác trong giải thuật.
  - Chương trình được tạo thành từ dãy các câu lệnh.
- Các câu lệnh trong C, được kết thúc bởi dấu chấm phẩy (;)
  - Chú ý: lệnh khối (sẽ học sau) không cần kết thúc bằng dấu ;

Phân loại

- Câu lệnh đơn:
  - Những câu lệnh không chứa câu lệnh khác.
    - Ví dụ: Phép gán, gọi hàm, vào/ra dữ liệu
- Các câu lệnh phức:
  - Những câu lệnh chứa câu lệnh khác.
    - Ví dụ: Lệnh khối (Tập các lệnh đơn nhóm lại với nhau và đặt trong cặp ngoặc nhọn « { } »)
  - Các lệnh điều khiển cấu trúc chương trình
    - Ví dụ: Lệnh rẽ nhánh, lệnh lặp..

10. Chú thích (comment)

- Lời mô tả, giải thích vắn tắt cho một câu lệnh, một đoạn chương trình hoặc cả chương trình
  - Giúp việc đọc hiểu chương trình dễ dàng hơn
  - Chú thích không phải là câu lệnh $\Rightarrow$ không ảnh hưởng tới chương trình
    - Khi gặp chú thích, trình biên dịch sẽ bỏ qua
- Cách viết chú thích
  - Chú thích một dòng: sử dụng //
  - Chú thích nhiều dòng: sử dụng /* và */

Chương 1: Tổng quan về ngôn ngữ C

1.1. Lịch sử phát triển của ngôn ngữ C

1.2. Các phần tử cơ bản của ngôn ngữ C

1.3. Cấu trúc cơ bản của chương trình C

1.4. Biên dịch chương trình C

Các phần cơ bản

Khai báo các tệp tiêu đề

#include

Khai báo các đối tượng toàn cục

- Định nghĩa kiểu dữ liệu mới
- Các biến, hằng
- Các nguyên mẫu hàm (prototype)

Định nghĩa hàm main()

{

}

Định nghĩa các hàm đã khai báo nguyên mẫu

1. Khai báo các tệp tiêu đề

- Liệt kê danh sách thư viện sẽ được sử dụng trong chương trình
  - Các hàm của C đều thuộc một thư viện nào đó
  - Nếu không khai báo thư viện, trình biên dịch sẽ không hiểu được hàm (có thể báo lỗi)
- Cách thức (cú pháp) khai báo
  - #include <ThuVien.h>
    - Thư viện phải nằm trong thư mục chứa các header file
    - Ví dụ: #include <stdio.h>
  - #include “ThuVien.h”
    - Thư viện nằm trong thư mục hiện tại

2. Khai báo các đối tượng toàn cục

- Các đối tượng toàn cục có phạm vi sử dụng trong toàn bộ chương trình
  - Các kiểu dữ liệu mới
  - Các hằng, biến
  - Các nguyên hàm
- Tuân theo nguyên tắc khai báo đối tượng

2. Khai báo các đối tượng toàn cục

- Định nghĩa kiểu dữ liệu
  - Cú pháp: typedef <ĐịnhNghĩaKiểu> <Tên kiểu>
  - Ví dụ: typedef unsigned char byte;

    typedef struct {float re, im;} complex;
- Khai báo hằng
  - const float PI = 3.1415;
  - #define Max 50
- Khai báo biến
  - int N;
  - float Delta, x1, x2;

2. Khai báo các đối tượng toàn cục (tiếp)

- Khai báo các nguyên mẫu hàm
- Khai báo thông tin về các hàm của người dùng sẽ được sử dụng trong chương trình
  - Tên hàm
  - Danh sách các kiểu tham số sẽ truyền vào
  - Kiểu dữ liệu trả về
- Ví dụ

```
float DienTichTamGiac(float a, float b, float c);
int getMax(int Arr []);
void swap(int * a, int  * b);
void swap(int *, int  *);
```

Có thể bỏ tên tham số

3. Định nghĩa hàm main()

- Bắt buộc phải có
- Là hàm đặc biệt trong C, đánh dấu điểm bắt đầu của mọi chương trình C
  - Khi thực hiện một chương trình C, hệ thống sẽ gọi tới hàm main đầu tiên, sau đó sẽ thực hiện lần lượt các câu lệnh (bao gồm cả lời gọi tới các hàm khác) nằm trong hàm main()
- Cú pháp
  - void main(){….}
  - void main(int argc, char * argv[ ]){….}
  - **int main(){….; return 0;}**
  - **int main(int argc, char * argv[ ]){….; return 0;}**

4. Định nghĩa các hàm đã khai báo

- Định nghĩa các hàm đã khai báo ở phần 3 (Phần khai báo nguyên mẫu - prototype)
  - Phần khai báo nguyên mẫu mới chỉ khai báo các thông tin cơ bản về hàm, chưa xác định rõ hàm hoạt động như thế nào
- Ví dụ

```
float DienTichTamGiac(float a, float b, float c){
    float p = (a+b+c)/2;
    return sqrt(p*(p-a)*(p-b)*(p-c));
}
```

$$S_{\Delta} = \sqrt{p(p-a)(p-b)(p-c)}$$

Chú ý

- Các phần không bắt buộc phải theo đúng thứ tự
- Nguyên tắc:
  - Mọi đối tượng cần phải được khai báo trước khi sử dụng
- Khi định nghĩa hàm được đặt trước hàm main() thì không cần khai báo nguyên mẫu hàm

Chương trình đầu tiên: Hello world!

```
1.  #include <stdio.h>
2.  int main() { //Không cần tham số dòng lệnh
3.      printf("Hello world! \n");
4.      return 0;  //Trả về giá trị 0
5.  }
```

- Khai báo thư viện stdio.h, đây là thư viện vào ra chuẩn (standard input output) chứa khai báo hàm printf
- Điểm bắt đầu thực hiện của chương trình. Máy tính thực hiện các câu lệnh nằm trong cặp ngoặc {} của main()
- Hàm printf in ra một xâu. Chú ý dấu xuống dòng (\n)
- Hàm main trả về giá trị 0 (0 thường dùng để thể hiện chương trình hoạt động bình thường, không có lỗi).

Chương 1: Tổng quan về ngôn ngữ C

1.1. Lịch sử phát triển của ngôn ngữ C

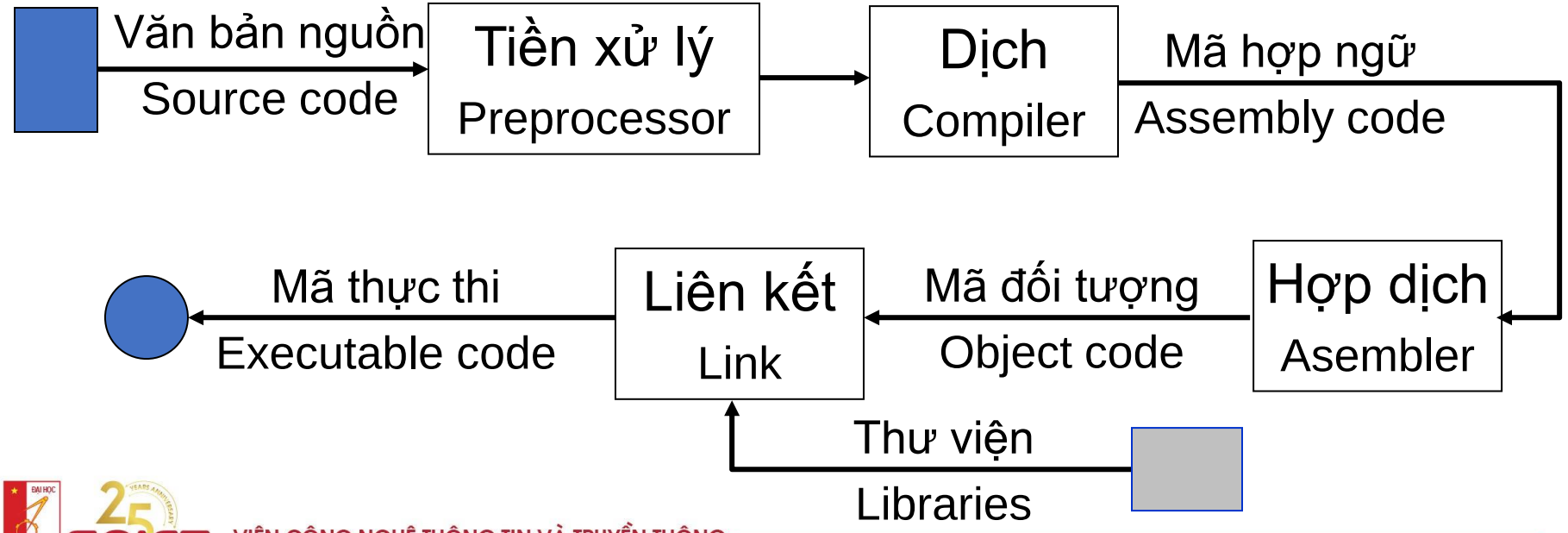
1.2. Các phần tử cơ bản của ngôn ngữ C

1.3. Cấu trúc cơ bản của chương trình C

1.4. Biên dịch chương trình C

Biên dịch chương trình

- Chương trình được viết bằng ngôn ngữ bậc cao phải được dịch ra mã máy để thực thi
  - Công việc dịch được thực hiện bởi trình biên dịch (compiler)
- Các giai đoạn dịch chương trình



Các công cụ lập trình

- Compiler:
  - Công cụ biên dịch mã nguồn thành chương trình
  - Visual C, gcc…
- IDE: Integrated Development Environment
  - Cung cấp giao diện giữa lập trình viên và compiler
  - Thực hiện tính năng: biên dịch, thực thi, gỡ lỗi…
  - MS Visual Studio, Eclipse…

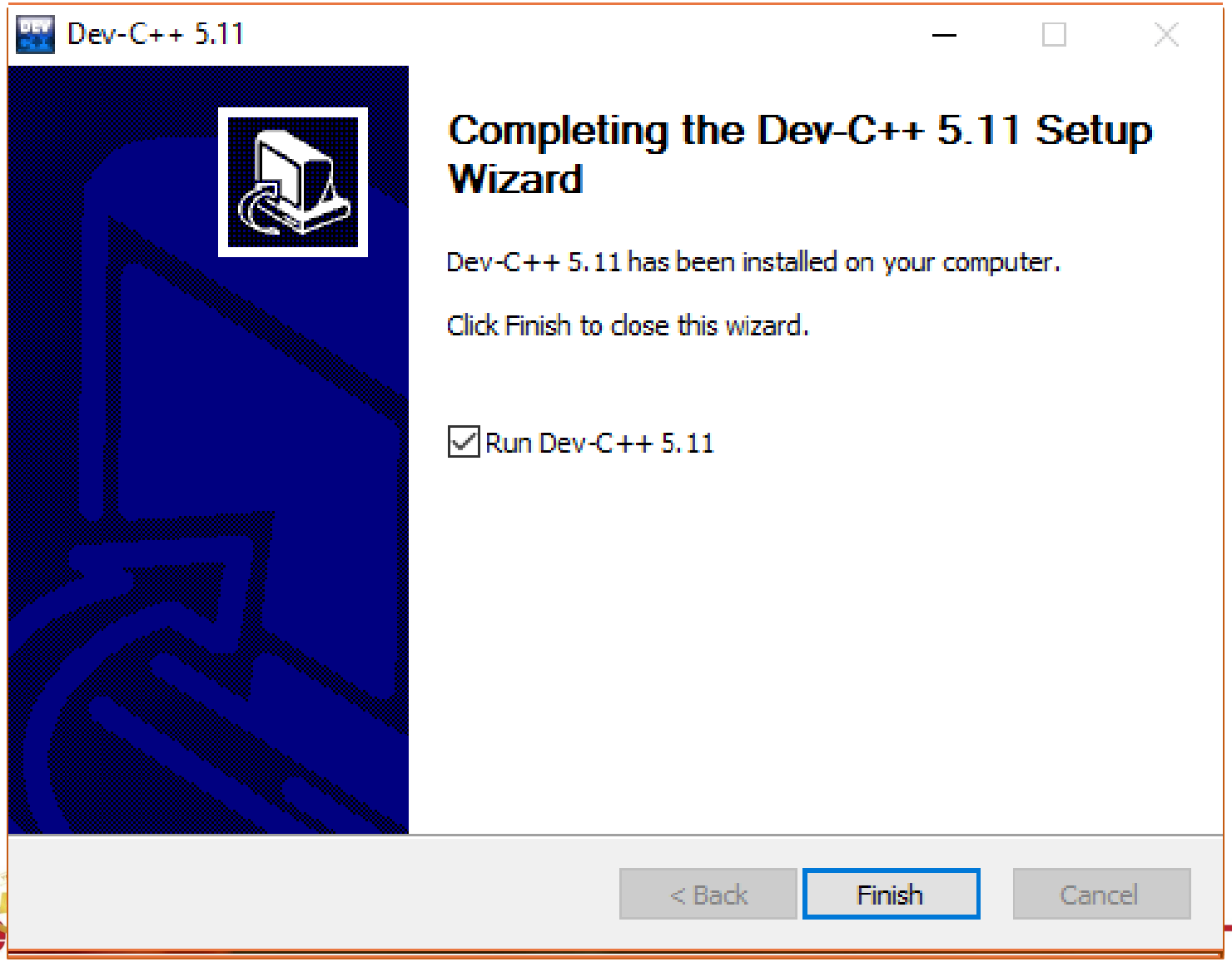
Dev-Cpp Ver 4.9.9.2 hoặc Dev-Cpp 5.1.1

Cài đặt Dev-Cpp

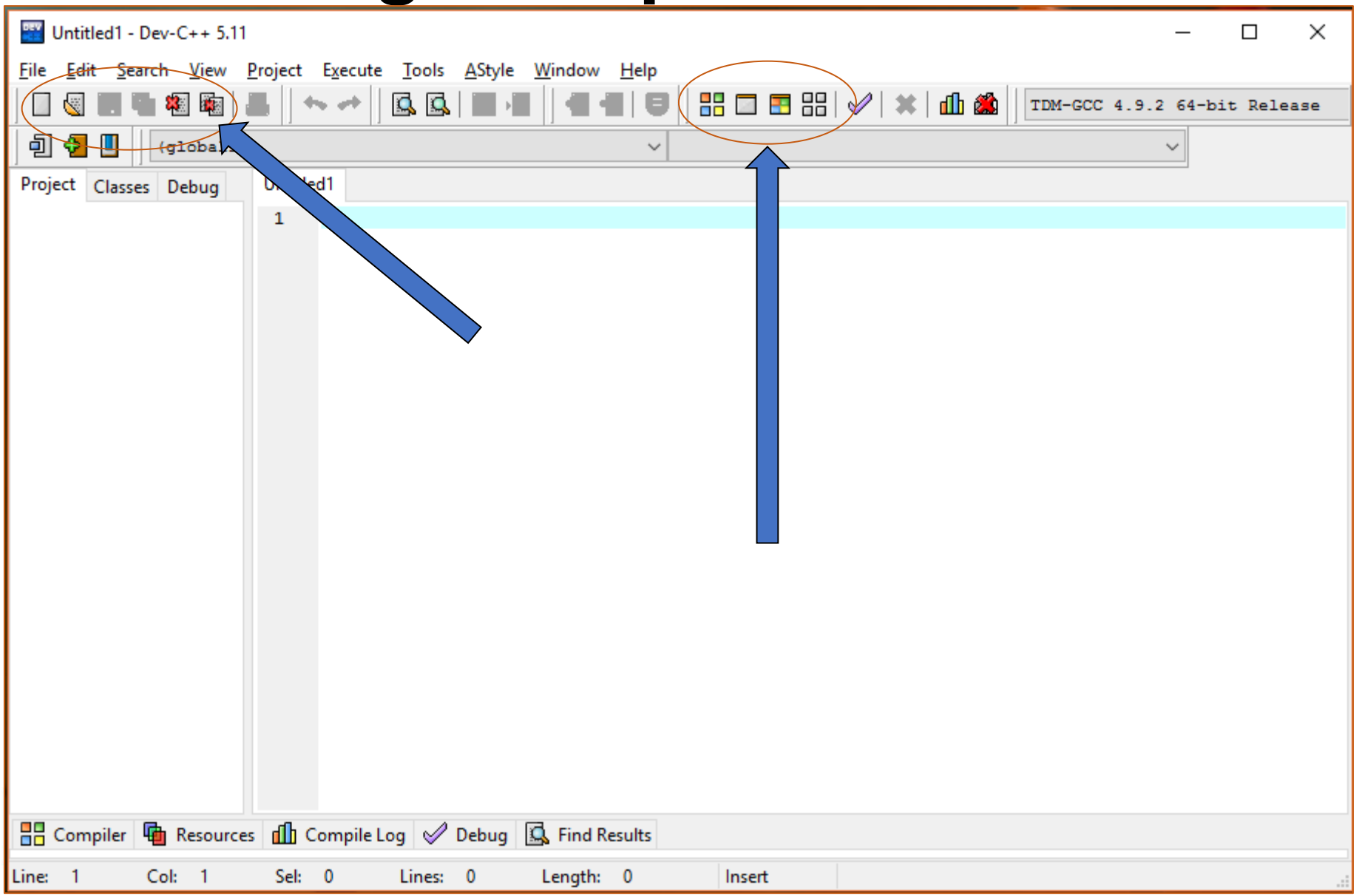
- Tải Dev-C++
  - Google ⇒ Dev-C++
  - Tìm đến phiên bản thích hợp, thực hiện tải về
    - Dev-Cpp 5.11 TDM-GCC 4.9.2 Setup.exe
    - Devcpp_v4_9_9_2_setup.exe

- Thực thi file tải về, làm theo hướng dẫn

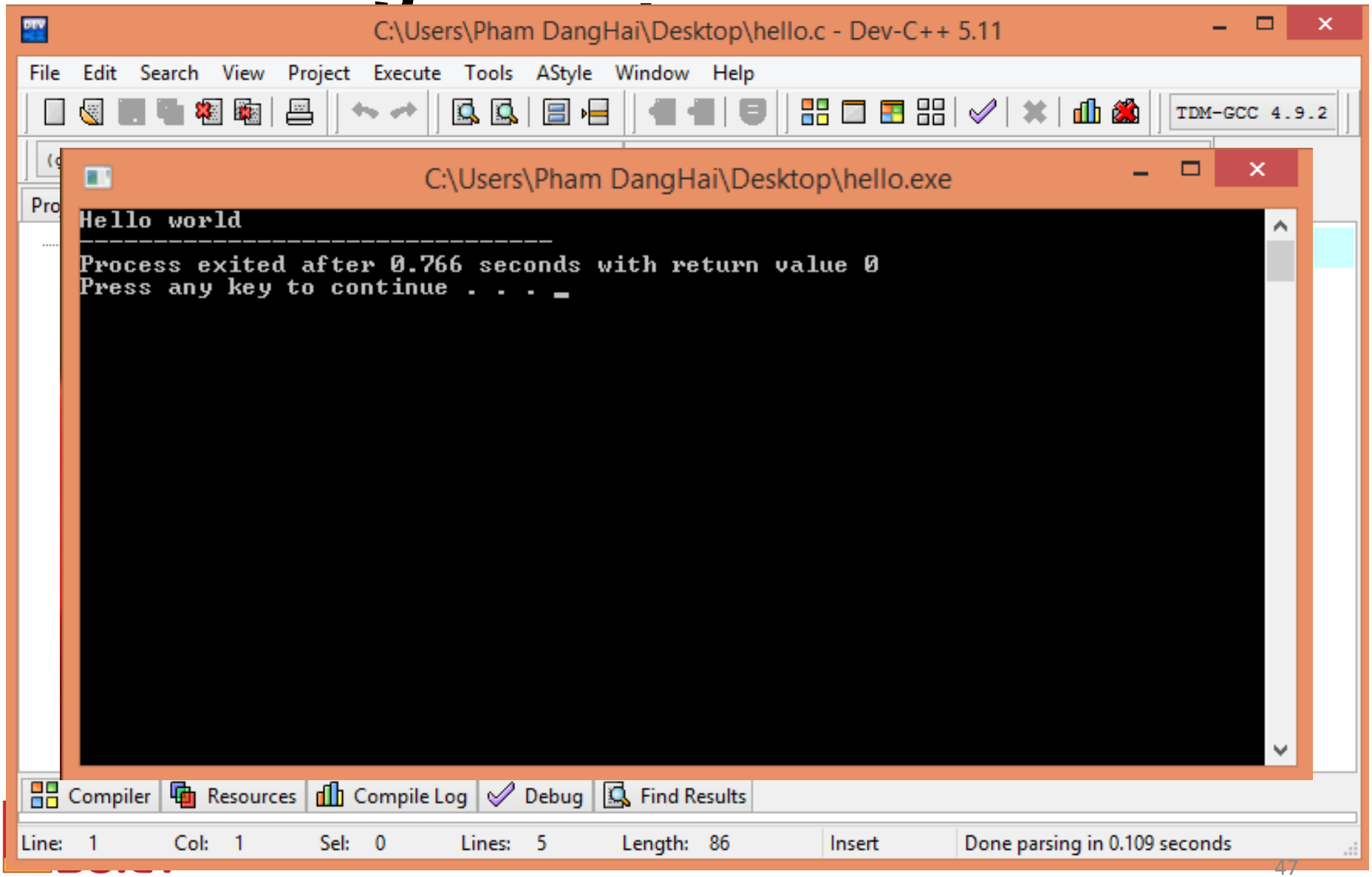
Cài đặt Dev-Cpp



Màn hình giao diện DEV-C++



Màn hình giao diện DEV-C++



Tóm tắt chương 1

- Lịch sử phát triển của ngôn ngữ C
- Các phần tử cơ bản của ngôn ngữ C
  - 10 phần tử cơ bản
- Cấu trúc cơ bản của chương trình C
  - 4 phần
- Thực hiện chương trình C với Dev-C++