

Nội dung môn học

- **Chương 1: Các khái niệm cơ bản của ngôn ngữ C**
- Chương 2: Các lện điều khiển
- Chương 3: Con trỏ và mảng
- Chương 4: Hàm
- Chương 5: Kiểu dữ liệu có cấu trúc Struct
- Chương 6: Đồ họa
- Chương 7: Thao tác vào ra với tệp

quangchieu.ta@gmail.com

CHƯƠNG 1: CÁC YẾU TỐ CƠ BẢN CỦA NGÔN NGỮ LẬP TRÌNH C

1.1. LỊCH SỬ HÌNH THÀNH VÀ PHÁT TRIỂN

- ✓ Ngôn ngữ C do Brian W. Kernighan và Dennis M. Ritchie phát triển vào đầu những năm 70.
- ✓ 1978, cuốn sách "The C programming langguage" của hai tác giả trên được phổ biến rộng rãi.
- ✓ 1983, Viện tiêu chuẩn Quốc gia Mỹ (ANSI) đề xuất chuẩn cho C.
- ✓ 1988, chuẩn ANSI C được ban hành. Hiện có các chương trình dịch C như: TURBO C, BORLAND C, C Free, VISUAL C ...

quangchieu.ta@gmail.com

1.2. CÁC TÍNH CHẤT CỦA C

- C là ngôn ngữ vạn năng được dùng để viết các hệ điều hành, các chương trình ứng dụng như quản lý văn bản, cơ sở dữ liệu...
- C có mức độ thích nghi cao, C được sử dụng rộng rãi.
- C có các cấu trúc điều khiển như: nhóm tuần tự các câu lệnh; lựa chọn quyết định (if, switch); chu trình (for, while, do...while).
- C cung cấp con trỏ và khả năng định địa chỉ số học.
- C cho phép hàm được gọi đệ quy.
- C tương đối thoải mái trong chuyển đổi dữ liệu.
- C không đưa ra các phép toán xử lý trực tiếp các đối tượng như xâu ký tự hoặc mảng.

1.3. TỪ VỰNG

1.3.1. Tập ký tự của ngôn ngữ C

- 26 chữ cái hoa: A, B, ... , X, Y, Z
- 26 chữ cái thường: a, b, ..., x, y, z
- 10 chữ số : 0, 1, 2, ... , 9
- Các ký hiệu toán học: +, -, *, /, =, (,)
- Ký tự gạch nối: _ (chú ý phân biệt với dấu trừ -)
- Các ký tự đặc biệt như: [ ] { } , . ; : & # % $
- Dấu cách: là một khoảng trống dùng để tách các từ.

1.3.2. Từ khóa

- Là những từ dành riêng cho ngôn ngữ C, chúng có ý nghĩa xác định và được sử dụng để khai báo các kiểu dữ liệu, để viết các toán tử , các câu lệnh...
- ***Chú ý:***
- Không dùng từ khóa để đặt tên cho các hằng, biến, mảng, hàm.
- Từ khóa phải được viết bằng chữ thường.

1.3.3. Tên

- Là dãy ký tự (chữ cái, chữ số, dấu gạch nối), ký tự đầu tiên của tên phải là chữ cái hoặc dấu gạch nối.
- Độ dài cực đại của tên mặc định là 32.
- ***Chú ý***: Trong các tên, chữ hoa và chữ thường được xem là khác nhau. *Ví dụ*: tên AB là khác với tên ab
- Trong C, thường dùng chữ hoa để đặt tên cho các hằng, dùng chữ thường để đặt tên cho các đại lượng khác như biến, mảng, hàm, cấu trúc…

CHƯƠNG 1: CÁC YẾU TỐ CƠ BẢN CỦA NGÔN NGỮ LẬP TRÌNH C

1.3.4. Các kiểu dữ liệu

Tên kiểu	ý nghĩa	Phạm vi	Kích thước
unsigned char	Ký tự	0 ÷ 255	1 byte
char	Ký tự	-128 ÷ 127	1 byte
unsigned int	Số nguyên	0 ÷ 65535	2 byte
int	Số nguyên	-32768 ÷ 32767	2 byte
unsigned long	Số nguyên	0 ÷ 4294967295	4 byte
long	Số nguyên	-2147483648 ÷ 2147483647	4 byte
float	Số thực dấu chấm động, độ chính xác đơn	±3.4E-38 ÷ ±3.4E+38	4 byte
double	Số thực dấu chấm động, độ chính xác kép	±1.7E-308 ÷ ±1.7E+308	8 byte
long double	Số thực dấu chấm động, độ chính xác lớn và kép	±3.4E-4932 ÷ ±1.1E+4932	10 byte

Kiểu void là kiểu không giá trị dùng để biểu diễn kết quả của hàm hay của con trỏ.

1.3.5. Hằng

- Hằng là đại lượng có giá trị không đổi khi thực hiện chương trình.
- Có hai cách khai báo hằng trong ngôn ngữ C:
- (1) Khai báo hằng dùng từ khóa const:
- Cú pháp:

```
const <kiểu dữ liệu>  < tên hằng>  = <giá trị> ;
```

- VD: const int MAX = 100 ;
- (2) Khai báo hằng dùng toán tử #define:
- Cú pháp:

```
#define  <tên hằng>  <giá trị>
```

- VD: #define MAX 100

Các loại hằng:

(1) Hằng thực:

Cách 1: Dạng thập phân (dạng dấu chấm tĩnh), VD: 1.23

Cách 2: Dạng khoa học (dạng dấu chấm động), VD: 0.12E3, hoặc 1.2E2, hoặc 12.E1 v.v...

(2) Hằng nguyên kiểu int: Là số nguyên có giá trị trong khoảng từ -32768 đến 32767

(3) Hằng nguyên kiểu long: được viết theo cách thêm L hoặc l vào đuôi, ví dụ: -48935L, -48935l.

(4) Hằng nguyên hệ 8: được viết theo cách:

$0c_1c_2c_3...$

Bắt đầu bằng số 0, còn c_i là một số nguyên trong khoảng từ 0 đến 7.

(5) Hằng nguyên hệ 16: được viết theo cách:

$0xc_1c_2c_3...$ hoặc $0Xc_1c_2c_3...$

Bắt đầu bằng số 0 và chữ x (hoặc X), còn c_i là một chữ số hệ 16 (0,1,...,9,a,b,c,d,e,f).

(6) Hằng ký tự: là một ký tự được viết trong 2 dấu nháy đơn,

ví dụ: 'a'. Giá trị của 'a' chính là mã ASCII của chữ "a". Hằng ký tự có thể tham gia vào các phép toán như mọi số nguyên khác.

Các hằng ký tự có cách viết đặc biệt:

'\n'	chuyển dòng mới				
'\t'	Tab				
'\0'	null	'\''	nháy đơn '	'\"'	nháy kép "
'\\'	gạch xiên \	'\b'	xoá lùi		
'\r'	CR về đầu dòng	'\f'	LF xuống dòng		

(7) Hằng xâu ký tự: Là một dãy ký tự bất kỳ đặt trong 2 dấu nháy kép, ví dụ: "Ha noi - Viet nam"

"" // đây là xâu rỗng

Một hằng xâu ký tự được lưu trữ trong bộ nhớ với tận cùng bằng một ký tự NULL ('\0')

CHƯƠNG 1: CÁC YẾU TỐ CƠ BẢN CỦA NGÔN NGỮ LẬP TRÌNH C

1.3.6. Biến

Biến là đại lượng có giá trị thay đổi trong chương trình.

Khai báo biến:

```
<kiểu dữ liệu> <danh sách các biến>;
```

VD:

```
int  n , i;
float  x , y;
```

- Các khai báo biến cần đặt ngay sau dấu "{" đầu tiên của thân hàm và cần đứng trước các lệnh khác.

- Có thể vừa khai báo vừa khởi tạo giá trị đầu cho biến,

- ví dụ: `int  a , b = 20 , c , d = 40 ;`

- Lấy địa chỉ của biến: Mỗi biến được cấp phát một vùng nhớ gồm một số byte liên tiếp, số hiệu của byte đầu tiên chính là địa chỉ của biến. Để nhận địa chỉ của biến ta dùng toán tử & như sau:

```
& <biến>
```

1.3.7. Mảng

Mảng là một tập hợp các biến (hay phần tử) có cùng một kiểu dữ liệu và có chung một tên, các phần tử phân biệt với nhau bởi chỉ số.

Khai báo mảng gồm:

- Kiểu mảng (ví dụ: int, float, double ...)
- Tên mảng.
- Số chiều và kích thước mỗi chiều.

Ví dụ:

```
int  a[10] ;
float  y[3][3] ;
```

Chú ý:

- Lấy địa chỉ phần tử mảng 1 chiều: `&a[i]`
- Không chấp nhận phép lấy địa chỉ phần tử mảng hai chiều, nghĩa là không thể viết `&a[i][j]`
- Địa chỉ phần tử đầu của mảng chính là tên mảng: `a = &a[0]`

1.3.8. Định nghĩa kiểu bằng typedef

Đặt từ khóa **typedef** vào trước một khai báo thông thường, khi đó tên dữ liệu (biến, mảng, cấu trúc ...) trở thành tên kiểu dữ liệu.

Ví dụ:

```
typedef  int  vecto [100] ;
typedef  float  matran [50][50] ;
```

Sau này ta có thể dùng các khai báo:

```
vecto  vt1, vt2 ;
matran  a , b , c ;
```

1.4. Biểu thức

1.4.1. Khái niệm

Biểu thức là một sự kết hợp giữa các phép toán và các toán hạng để diễn đạt một công thức toán học nào đó.

(Toán hạng: có thể là hằng, biến, biểu thức).

Biểu thức được sử dụng:

- Làm vế phải của lệnh gán
- Làm tham số thực sự của hàm
- Làm chỉ số của phần tử mảng
- Trong các câu lệnh như if, for, while, do ... while
- Trong các biểu thức phức tạp hơn

Ví dụ:

```
2.0 * (x + y)
sin(x / 3) * (x / y) + exp(z)
(delta_x > 0.0) && (delta_y > 0.0)
```


1.4.2. Các phép toán số học

Phép toán	ý nghĩa	Ví dụ
-	Đổi dấu	-a
+	Cộng	1.25 + a
-	Trừ	a – b
*	Nhân	x * y
/	Chia	m / 2.0
%	Lấy số dư của phép chia 2 số nguyên	15 % 4

Chú ý: Phép chia / cho kết quả nguyên nếu cả 2 toán hạng có kiểu số nguyên

1.4.3. Các phép toán quan hệ: Cho kết quả là 0 (sai), hoặc ≠ 0 (đúng)

Phép toán	ý nghĩa	Ví dụ
>	So sánh lớn hơn	a > b
>=	So sánh lớn hơn hoặc bằng	x >= y
<	So sánh nhỏ hơn	i < n+1
<=	So sánh nhỏ hơn hoặc bằng	k <= i - 1
==	So sánh bằng nhau	i == j
!=	So sánh khác nhau	x != y

1.4.4. Các phép toán logic: Cho kết quả là 0 (sai), hoặc ≠ 0 (đúng)

Phép toán	ý nghĩa	Ví dụ
&&	"và" logic	(ch >= 0) && (ch <= 9)
||	"hoặc" logic	(ch = 'c') || (ch = 'C')
!	"phủ định" một ngôi	!a

1.4.5. Câu lệnh gán và biểu thức gán

- Câu lệnh gán có dạng:

```
v = e ;
```

Trong đó v là biến, e là biểu thức.

Một số ký pháp đặc biệt của phép gán:

Dạng viết thông thường	Dạng viết thu gọn	ý nghĩa
i = i + <biểu thức> ;	i += <biểu thức> ;	Tự cộng
i = i - <biểu thức> ;	i -= <biểu thức> ;	Tự trừ
i = i * <biểu thức> ;	i *= <biểu thức> ;	Tự nhân
i = i / <biểu thức> ;	i /= <biểu thức> ;	Tự chia
i = i % <biểu thức> ;	i %= <biểu thức> ;	Tự lấy d

Ví dụ:

```
- i = 1 ;
- thetich = 4.0 * M_PI * r * r * r / 3.0 ;
- a = b = c = d = 100 ;
```

Tương đương với `a = (b = (c = (d = 100))) ;`

(các giá trị của a, b, c, d đều là 100).

```
- m = (i = 3) * ( j = 5 ) ;
```

Sẽ cho kết quả m = 15

CHƯƠNG 1: CÁC YẾU TỐ CƠ BẢN CỦA NGÔN NGỮ LẬP TRÌNH C

1.4.6. Biểu thức điều kiện

Biểu thức điều kiện có dạng:

<e1> ? <e2> : <e3> ;

Trong đó e1, e2, e3 là các biểu thức.

Giá trị của biểu thức điều kiện bằng <e2> nếu <e1> khác 0, và bằng <e3> nếu <e1> là 0.
Kiểu của biểu thức điều kiện là kiểu cao nhất của <e2> và <e3>.

Ví dụ:

max2 = (a > b) ? a : b ;

Cho giá trị lớn nhất trong 2 số a,b.

1.4.7. Các phép toán tăng, giảm một đơn vị

Toán tử	Dạng tiền tố	Dạng hậu tố
Tăng 1 đơn vị	++ n	n ++
Giảm 1 đơn vị	-- n	n --

Trong dạng tiền tố, thay đổi giá trị của biến trước khi sử dụng biến để tính toán.

Trong dạng hậu tố, giá trị cũ của biến được sử dụng để tính toán, rồi sau đó biến mới được thay đổi giá trị.

Ví dụ:

Giả sử i = 3, j = 15, ta xét các trường hợp:

Phép toán	Tương đương	Kết quả
i = ++j	j = j + 1 ; i = j ;	i = 16 và j = 16
i = j++	i = j ; j = j + 1 ;	i = 15 và j = 16
i++ ;	i = i + 1 ;	i = 4
j = ++i + 5 ;	i = i + 1 ; j = i + 5 ;	i = 4 và j = 9
j = i ++ + 5	j = i + 5 ; i = i + 1 ;	i = 4 và j = 8

1.4.8. Chuyển đổi kiểu giá trị

- Chuyển đổi kiểu tự động trong biểu thức

Chuyển đổi tự động tuân theo sơ đồ sau:

char → int → long → float → double → long double

- Chuyển đổi kiểu thông qua phép gán

VD: **float x; int n; …; x = n;**

- Dùng phép ép kiểu

(<kiểu>) <biểu thức>, VD: (float) n;

- Dùng các hàm chuyển đổi kiểu:

Hàm int **atoi(char *s)** : chuyển xâu ký tự s sang số nguyên kiểu int

Hàm long **atol(char *s)** : chuyển xâu s sang số nguyên dài kiểu long

Hàm double **atof(char *s)** : chuyển xâu s sang số thực kiểu double

Hàm **itoa(<số nguyên int>, <xâu>, <hệ cơ số>)**

Hàm **ltoa(<số nguyên long>, <xâu>, <hệ cơ số>)**

Hàm **ultoa(<số nguyên unsigned long >, <xâu>, <hệ cơ số>)**

1.4.9. Các hàm toán học

Hàm	ý nghĩa
int abs (int x) long labs (long x) double fabs (double x)	Hàm cho giá trị tuyệt đối |x|
double acos (double x)	Hàm tính arccos x
double asin (double x)	Hàm tính arcsin x
double atan (double x)	Hàm tính arctg x
double cos (double x)	Hàm cos x
double sin (double x)	Hàm sin x
double tan (double x)	Hàm tg x

1.4.9. Các hàm toán học (tiếp)

Hàm	ý nghĩa
double ceil (double x)	Hàm xác định số nguyên (kiểu double) bé nhất trên x
double floor (double x)	Hàm xác định số nguyên (kiểu double) lớn nhất dới x
double exp (double x)	Hàm tính e^x
double log (double x)	Hàm tính ln x
double log10 (double x)	Hàm tính lg x
double pow (double y ,double x)	Hàm tính y^x
double sqrt (double x)	Hàm tính

1.5. Nhập xuất dữ liệu

1.5.1. Hàm xuất printf ( )

```
printf(< xâu quy cách >,< Danh sách đối số>);
```

Trong đó:

< xâu quy cách > là một xâu ký tự chứa các định dạng xác định cách thức hiển thị dữ liệu.

<Danh sách đối số> (nếu có) sẽ là các giá trị của hằng, biến, biểu thức. Các giá trị trong danh sách này cách nhau bởi dấu phẩy ",". Số lượng các giá trị trong danh sách này bằng số định dạng có mặt trong <xâu quy cách>.

Chú ý:

- <xâu quy cách> có thể không chứa một định dạng nào, ví dụ: *printf("\nXau dieu khien la mot hang xau ky tu");*
- <xâu quy cách> không được là một biến kiểu xâu, ví dụ không thể viết:

```
char  *xau = "Noi dung bien xau ky tu" ;
printf (xau) ;
```

Định dạng	áp dụng cho kiểu d liệu	Ghi chú
%d	int	Đối là một số nguyên kiểu int
%i	int	Đối là số hệ 10 có dấu (nếu giá trị âm)
%ld	long	Đối là một số nguyên kiểu long
%o	int	Đối là số hệ 8 không dấu
%u	int	Đối là số nguyên không dấu
%x	int	Đối số là số nguyên hệ 16 không dấu (không có Ox đứng trước)
%X	int	Số nguyên hệ 16 với các ch số là A B C D E F
%e, %E	float, double	Đối số được chuyển sang dạng thập phân dấu chấm động
%f, %lf	float, double	Đối số được chuyển sang dạng thập phân dấu chấm tĩnh
%g, %G	float, double	Đối số được chuyển sang dạng thập phân dấu chấm tĩnh hay động tùy thuộc vào loại nào ngắn hơn
%c	char	Đối số là một ký tự
%s	char *	Đối số là một xâu ký tự
\n		Xuống dòng mới
\t		Nhảy một đoạn Tab

quangchieu.ta@gmail.com

CHƯƠNG 1: CÁC YẾU TỐ CƠ BẢN CỦA NGÔN NGỮ LẬP TRÌNH C

- Lệnh printf ( ) với các định dạng có độ rộng:

Cách viết	ý nghĩa
%wd	In ra số nguyên trong khuôn có độ rộng là w
%wc	In ra ký tự trong khuôn có độ rộng là w
%w.pf	In ra số thực trong khuôn có độ rộng là w và có p ch số sau dấu chấm thập phân
%wx	In ra số nguyên hệ 16 trong khuôn có độ rộng là w
%ws	In ra xâu ký tự trong khuôn có độ rộng là w

quangchieu.ta@gmail.com

Ví dụ 1:

```
printf("Bai hoc C dau tien. \n"); /* Bai hoc C dau tien la chuoi ki tu, \n la ki tu dieu khien xuong dong */
```

Ví dụ 2:

giả sử có biến i=6 in ra màn hình giá trị của biến i ta viết

```
printf("so ban nhap la : %d.",i);
```

Ví dụ 3:

giả sử có biến a=4 b=5 in ra a+b,a,b ta có dòng code

```
printf("tong cua 2 so %d va %d la %d.",a,b,a+b);
```

Ví dụ 4:

sửa lại ví dụ 3 như sau

```
printf("Tong cua 2 so %5d va %3d la %1d . \n", a, b, a+b);
```


1.5.2. Hàm nhập scanf ( )

```
scanf (<xâu quy cách>,<Danh sách địa chỉ biến>);
```

Hàm này đọc các ký tự từ bàn phím, biến đổi chúng tương ứng theo khuôn dạng được xác định bằng các "định dạng" trong <xâu quy cách>, rồi lưu trữ kết qủa trong các đối số có địa chỉ trong phần <Danh sách địa chỉ >. Số lượng "định dạng" trong <xâu quy cách> cũng phải bằng số địa chỉ trong phần <Danh sách địa chỉ >.

Chú ý:

Để nhập đúng giá trị cho biến ký tự thi:

(i) phải sử dụng một lệnh scanf ( ) riêng biệt cho mỗi lần nhập ký tự.

(ii) Trước lệnh scanf ( ) cần đặt lệnh fflush (stdin) để làm sạch vùng bộ nhớ đệm *stdin.*

(iii) giá trị nhập vào biến xâu ký tự không được chứa dấu cách.

Định dạng	ý nghĩa
%d	Nhập vào một số nguyên, đối số tương ứng phải là một địa chỉ của một biến nguyên
%o	Nhập vào một số nguyên hệ 8, đối số tương ứng phải là một địa chỉ của một biến nguyên
%x	Nhập vào một số nguyên hệ 16, đối số tương ứng phải là một địa chỉ của một biến nguyên
%c	Nhập vào một ký tự, địa chỉ tương ứng phải là một địa chỉ của một biến ký tự
%s	Nhập vào một xâu ký tự, đối số tương ứng phải là một xâu ký tự
%f	Nhập vào một số thực float, đối số tương ứng phải là một địa chỉ của một biến thực
%ld	Nhập vào một số nguyên long
%lf	Nhập vào một số thực double

Ví dụ : Chương trình C sau minh họa cách sử dụng của hàm scanf() trong C:

```
#include <stdio.h>
int main()
{
   char str1[20], str2[30];
   printf("Nhap ten: ");
   scanf("%s", &str1);
   printf("Nhap so dien thoai va ngay sinh: ");
   scanf("%s", &str2);
   printf("Ten vua nhap: %s\n", str1);
   printf("So dien thoai va ngay sinh vua nhap: %s", str2);
   return(0);
}
```

1.5.3. Một số hàm nhập – xuất ký tự và xâu

a/Hàm **gets (char *s)**:

Nhập một xâu ký tự, trong đó s là con trỏ kiểu char trỏ tới vùng nhớ chứa dãy ký tự nhập vào.

b/ Hàm **getchar ( )**:

Nhận 1 ký tự từ bàn phím. Hàm trả về ký tự nhận được.

c/ Hàm **puts (const char *s)**:

Đưa chuỗi s lên màn hình.

d/ Hàm **putchar (int ch)**:

Đưa ký tự *ch* lên màn hình

e/ Hàm **getch ( )**:

Công dụng: Nếu bộ đệm rỗng thì máy tạm dừng, khi gõ 1 ký tự thì hàm nhận ngay (không cần bấm thêm Enter), ký tự vừa gõ không được hiện lên màn hình.

f/ Hàm **getche ( )**:

Hàm này làm việc giống như hàm getch( ), chỉ khác là cho hiện ký tự được gõ lên màn hình.

g/ Hàm putch(int ch):

Đưa ký tự *ch* lên màn hình, ký tự sẽ được hiển thị theo màu xác định trong hàm ***textcolor***.

1.5.4. Một số hàm về thao tác trên màn hình

a/ Hàm xoá màn hinh: clrscr( )

b/ Hàm di chuyển con trỏ màn hinh: gotoxy(x, y)

c/ Hàm wherex ( ): cho toạ độ x của con trỏ

d/ Hàm wherey ( ): cho toạ độ y của con trỏ

e/ Hàm textcolor (màu): đặt màu của van bản.

f/ Hàm textbackground (màu): đặt màu nền.

1.6. Một vài ví dụ về chương trinh C đơn giản

Ví dụ 1: Viết chương trình hiện lên màn hình dòng chư "TURBO C xin chao ban !".

```
#include <stdio.h> // sử dụng thư viện vào - ra chuẩn
#include <conio.h>  // sử dụng thư viện quản lý màn hình
void main()         //  hàm chính
{
   clrscr( );       // xóa màn hình
   gotoxy(30,12);    // đưa con trỏ đến toạ độ (30,12)
   printf("TURBO C xin chao ban !");   // in ra màn hinh
   getch( );        //  tạm dừng máy để xem kết quả
}
```


Ví dụ 2: Viết chương trinh nhập vào chiều dài và chiều rộng, sau đó tính chu vi và diện tích của hình chữ nhật.

```
#include <stdio.h>
#include <conio.h>
void main()
{
   float  dai, rong, chuvi, dientich;   // khai báo 4 biến thực
   clrscr( );
   printf("Nhap vao chieu dai cua hinh chu nhat: ");
   scanf("%f", &dai);
   printf("Nhap vao chieu rong cua hinh chu nhat: ");
   scanf("%f", &rong);
   chuvi = (dai+rong)*2;
   dientich = dai*rong;
   printf("\nChu vi = %10.3f\nDien tich = %10.3f", chuvi,
   dientich);
   getch();
}
```

1.7. Chạy một chương trình C trong TURBO C

Gồm các bước sau:

- Tạo tệp chương trình gốc có đuôi C (soạn thảo chương trinh).
- Dịch chương trình, tạo ra tệp chương trình thực hiện có đuôi EXE.
- Chạy chương trình

1.8. Hàm main( ):

Là một hàm đặc biệt, chương trinh C luôn luôn bắt đầu thực hiện tại điểm đầu của hàm này, có nghĩa là mọi chương trinh đều phải có một và chỉ một hàm *main ( )*, hàm này sẽ gọi các hàm khác để thực hiện.

1.9. Khai báo tệp tiêu đề

Khi sử dụng các hàm trong thư viện chuẩn, phải khai báo tệp tiêu đề (header file) chứa các hàm nguyên mẫu tương ứng của hàm đó, câu lệnh bắt đầu bằng *#include* theo sau là tên tệp tiêu đề. Ví dụ:

```
#include <stdio.h>
#include <conio.h>
```

1.10. Dòng chú thích, dấu chấm phẩy kết thúc lệnh

- Lời chú thích nằm giữa "/*" và "*/" . Nếu lời chú thích chỉ ở một dòng thi dùng “//”.
- Mỗi câu lệnh được kết thúc bằng dấu ";"

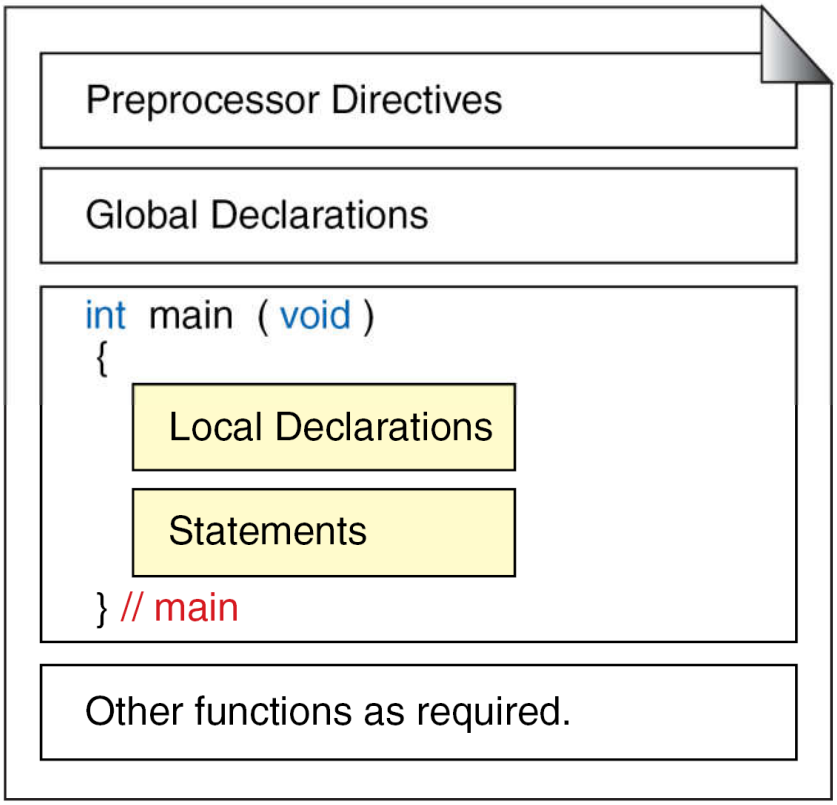

FIGURE 2-2 Structure of a C Program

quangchieu.ta@gmail.com

CHƯƠNG 1: CÁC YẾU TỐ CƠ BẢN CỦA NGÔN NGỮ LẬP TRÌNH C

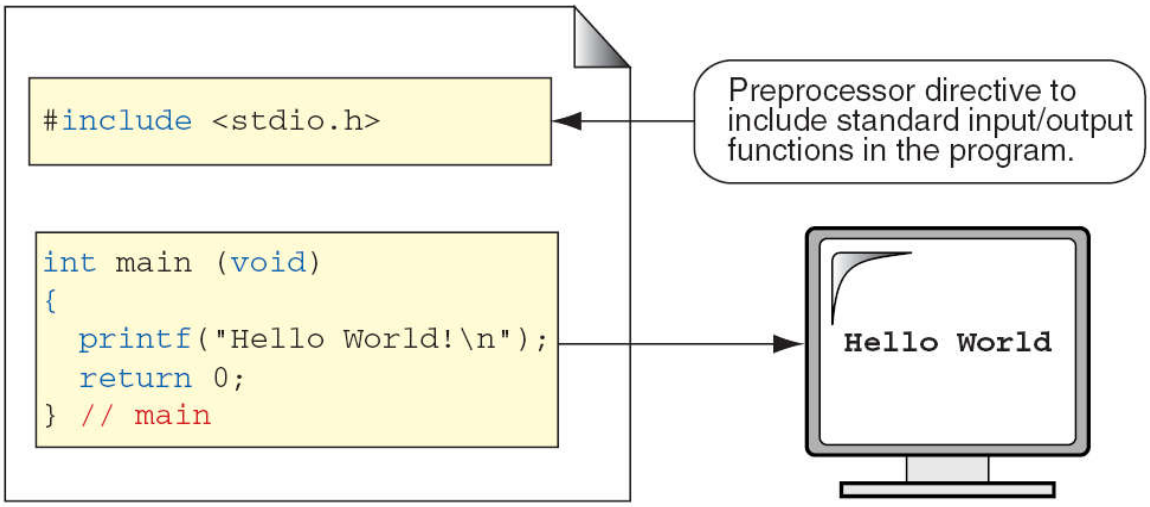

FIGURE 2-3 The Greeting Program

quangchieu.ta@gmail.com

```c
/* The greeting program. This program demonstrates
   some of the components of a simple C program.
      Written by:  your name here
      Date:        date program written
*/
#include <stdio.h>

int main (void)
{
// Local Declarations

// Statements

   printf("Hello World!\n");

   return 0;
} // main
```


PROGRAM 2-1 The Greeting Program

```c
/* The greeting program. This program demonstrates
   some of the components of a simple C program.
      Written by:  your name here
      Date:        date program written
*/
#include <stdio.h>

int main (void)
{
// Local Declarations

// Statements

   printf("Hello World!\n");

   return 0;
} // main
```