BACHKHOA EDUCATION APTECH
Làm trước học sau

TRƯỜNG ĐÀO TẠO LẬP TRÌNH VIÊN VÀ QUẢN TRỊ MẠNG QUỐC TẾ BACHKHOA-APTECH

BÀI 9: HÀM

MỤC TIÊU BÀI HỌC

- Tìm hiểu cách sử dụng hàm
- Tìm hiểu cấu trúc của hàm
- Khai báo hàm và các nguyên mẫu hàm
- Tìm hiểu các kiểu khác nhau của biến
- Hàm được gọi như thế nào
- Truyền bằng giá trị
- Truyền bằng tham chiếu
- Tìm hiểu về các qui tắc về phạm vi của hàm
- Các hàm trong các chương trình có nhiều tập tin
- Các lớp lưu trữ
- Con trỏ hàm

ĐỊNH NGHĨA HÀM

- ✓ Hàm là một đoạn chương trình thực hiện một tác vụ được định nghĩa cụ thể
- ✓ Các hàm được sử dụng để rút gọn cho một chuỗi các chỉ thị được thực hiện nhiều lần
- ✓ Hàm dễ viết và dễ hiểu
- ✓ Việc gỡ lỗi chương trình trở nên dễ dàng hơn khi cấu trúc của chương trình rõ ràng với hình thức lập trình theo module
- ✓ Chương trình cấu tạo từ các hàm cũng dễ dàng bảo trì, bởi vì sự sửa đổi khi có yêu cầu được giới hạn trong từng hàm của chương trình

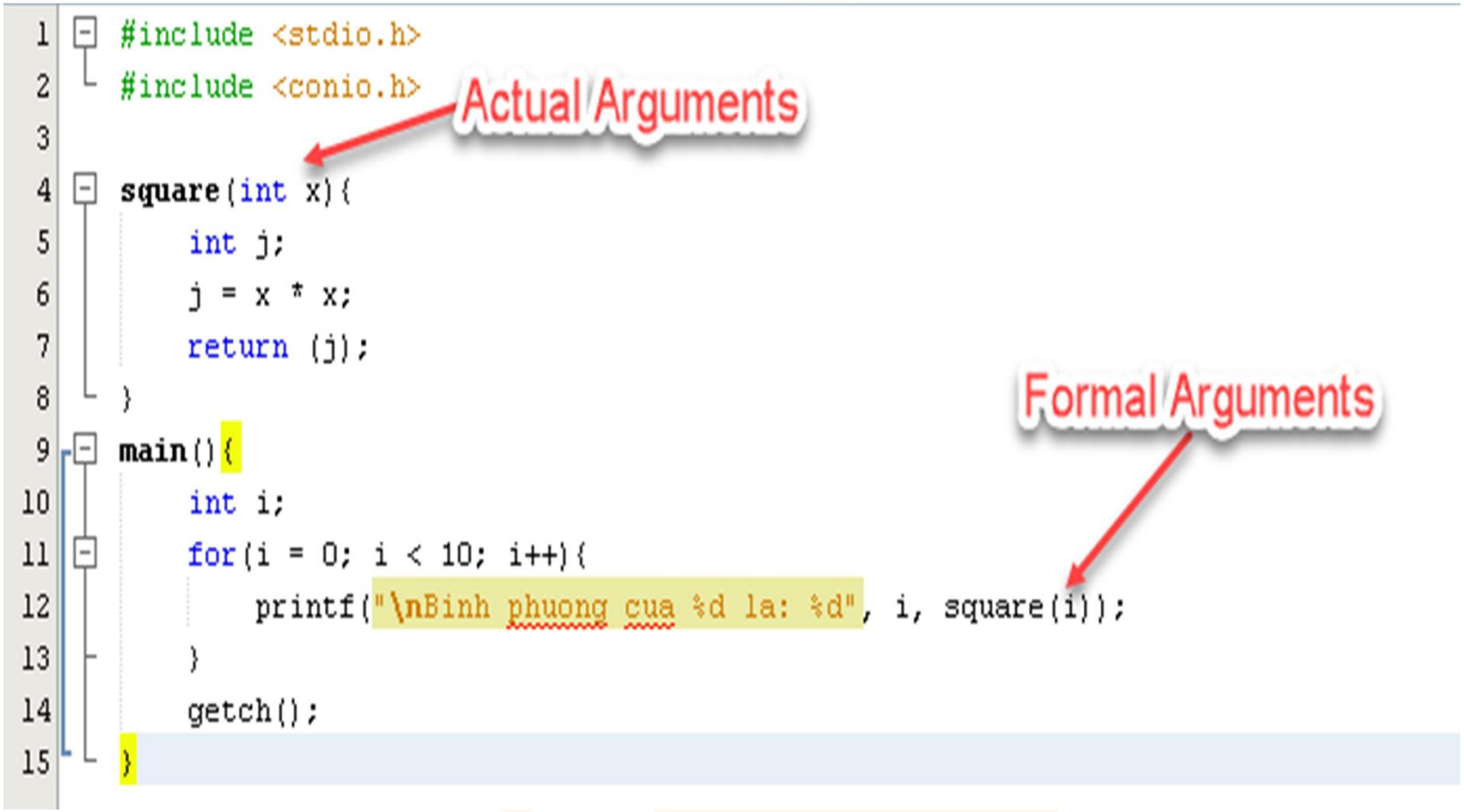
CẤU TRÚC HÀM

- ✓ Cú pháp tổng quát của một hàm trong C như sau:

```
type_specifier function_name(arguments)
{
        body of the function
}
```

- ✓ *type_specifier* xác định kiểu dữ liệu của giá trị mà hàm sẽ trả về.
- ✓ Một tên hàm hợp lệ được gán cho định danh của hàm
- ✓ Các đối số xuất hiện trong cặp dấu ngoặc () được gọi là các tham số hình thức.

CÁC ĐỐI SỐ CỦA HÀM




```
1  #include <stdio.h>
2  #include <conio.h>
3
4  square(int x){
5      int j;
6      j = x * x;
7      return (j);
8  }
9  main(){
10     int i;
11     for(i = 0; i < 10; i++){
12         printf("\nBinh phuong cua %d la: %d", i, square(i));
13     }
14     getch();
15 }
```

Chương trình tính bình phương của các số từ 1 đến 10

Dữ liệu được truyền từ hàm main() đến hàm square()

Hàm thao tác trên dữ liệu sử dụng các đối số

SỰ TRẢ VỀ CỦA MỘT HÀM

```
square(int x){
    int j;
    j = x * x;
    return (j);
}
```

- ✓ Lệnh return ngay lập tức chuyển điều khiển từ hàm trở về chương trình gọi.
- ✓ Giá trị đặt trong cặp dấu ngoặc () theo sau lệnh return được trả về cho chương trình gọi.

KIỂU DỮ LIỆU CỦA HÀM

```
type_specifier function_name (arguments)
{
      body of the function
}
```

- ✓ *type_specifier* không xuất hiện trước hàm squarer(), vì squarer() trả về một giá trị kiểu số nguyên int
- ✓ *type_specifier* là không bắt buộc nếu kiểu của giá trị trả về là một số nguyên hoặc nếu không có giá trị trả về
- ✓ Tuy nhiên, để tránh sự không nhất quán, một kiểu dữ liệu nên được xác định.

GỌI HÀM

- Dấu chấm phẩy được đặt cuối câu lệnh khi gọi hàm, nhưng không dùng cho định nghĩa hàm
- Cặp dấu ngoặc () là bắt buộc theo sau tên hàm, cho dù hàm có đối số hay không
- Nhiều nhất một giá trị được trả về
- Chương trình có thể có nhiều hơn một hàm
- Hàm gọi đến một hàm khác được gọi là hàm gọi
- Hàm đang được gọi đến được gọi là hàm được gọi

KHAI BÁO HÀM

- ✓ Việc khai báo hàm là bắt buộc khi hàm được sử dụng trước khi nó được định nghĩa
- ✓ Hàm address() được gọi trước khi nó
- ✓ được định nghĩa
- ✓ Một số trình biên dịch C sẽ thông báo
- ✓ lỗi nếu hàm không được khai báo trước khi gọi.
- ✓ Điều này còn được gọi là sự khai báo
- ✓ không tường minh

```
#include <stdio.h>
Main(){
        ...
        address()
        ...
}
address(){
        ...
}
```

NGUYÊN MẪU HÀM

- Xác định kiểu dữ liệu của các đối số

char abc(int x, nt y);

Thuận lợi :

Bất kỳ sự chuyển kiểu không hợp lệ giữa các đối số được dùng để gọi hàm và kiểu đã được định nghĩa cho các tham số của hàm sẽ được thông báo.

char noparam (void);

CÁC BIẾN

Biến cục bộ

- Được khai báo bên trong một hàm
- Được tạo tại điểm vào của một khối và bị hủy tại điểm ra khỏi khối đó.

Tham số hình thức

- Được khai báo trong định nghĩa hàm như là các tham số
- Hoạt động như một biến cục bộ bên trong một hàm

Biến toàn cục

- Được khai báo bên ngoài tất cả các hàm
- Lưu các giá trị tồn tại suốt thời gian thực thi của chương trình

LỚP LƯU TRỮ 1-2

Mỗi biến trong C có một tính chất được gọi là lớp lưu trữ

Lớp lưu trữ định nghĩa hai đặc tính của biến:

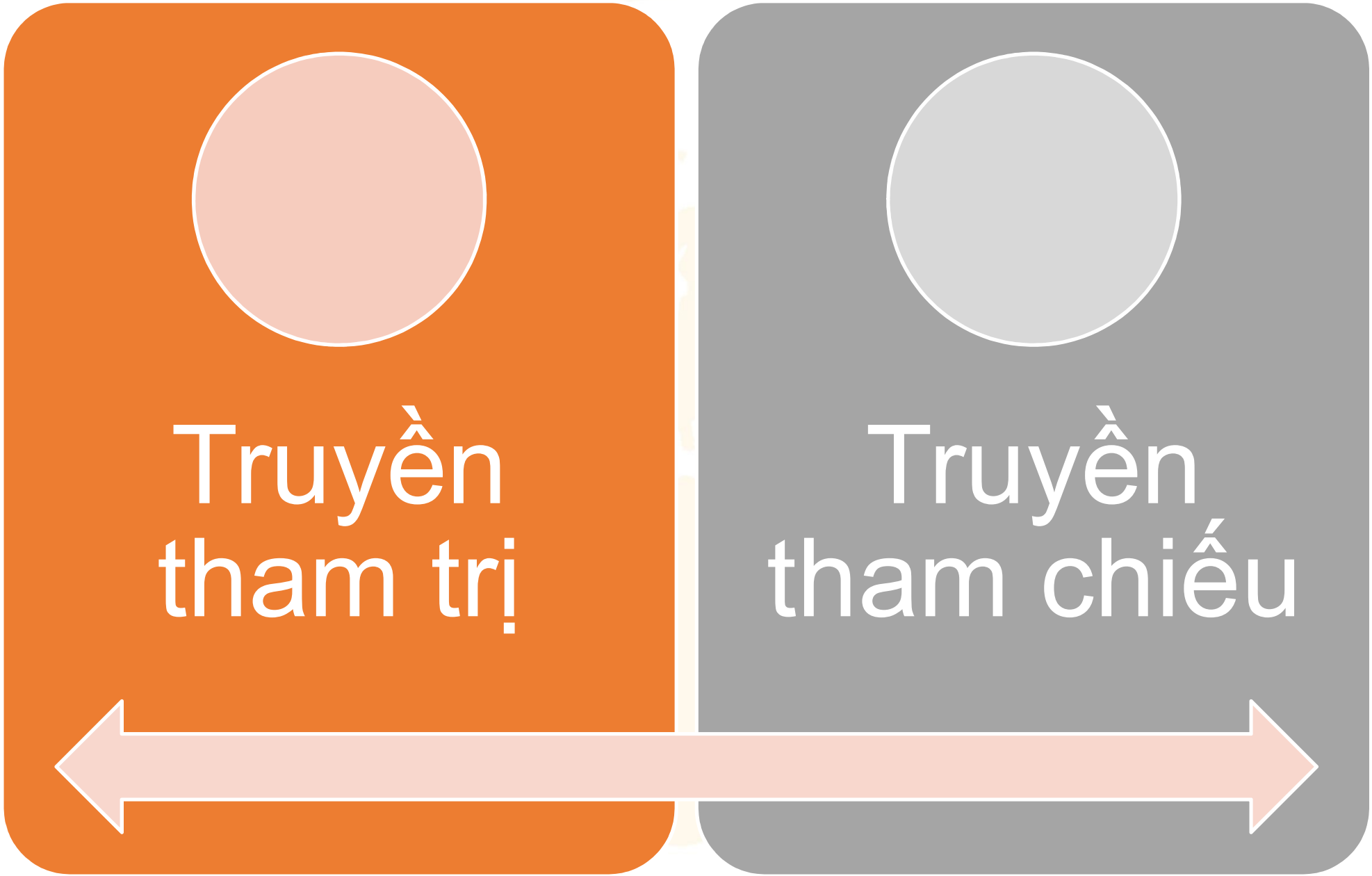
Thời gian sống của một biến là khoảng thời gian nó duy trì một giá trị xác định

Tầm vực của một biến xác định các phần của một chương trình có thể nhận ra biến đó

CÁC QUY LUẬT PHẠM VI CỦA HÀM

- ✓ Các qui luật phạm vi – là những qui luật quyết định một đoạn mã lệnh có thể truy xuất đến một đoạn mã lệnh hay dữ liệu khác hay không
- ✓ Mã lệnh bên trong một hàm là cục bộ với hàm đó
- ✓ Hai hàm có phạm vi khác nhau
- ✓ Hai hàm có cùng mức phạm vi
- ✓ Một hàm không thể được định nghĩa bên trong một hàm khác

GỌI HÀM



TRUYỀN BẰNG THAM TRỊ

- ✓ Mặc nhiên trong C, tất cả các đối số được truyền bằng giá trị
- ✓ Khi các đối số được truyền đến hàm được gọi, các giá trị được truyền thông qua các biến tạm
- ✓ Mọi sự thao tác chỉ được thực hiện trên các biến tạm
- ✓ Các đối số được gọi là truyền bằng giá trị khi giá trị của biến được truyền đến hàm được gọi và bất kỳ sự thay đổi trên giá trị này không ảnh hưởng đến giá trị gốc của biến được truyền

TRUYỀN BẰNG THAM CHIẾU

- ✓ Với truyền tham chiếu, hàm cho phép truy xuất đến địa chỉ thực trong bộ nhớ của đối số và vì vậy có thể thay đổi giá trị của các đối số của hàm gọi
- ✓ Định nghĩa
- ✓ **getstr(char *ptr_str, int *ptr_int);**
- ✓ Gọi
- ✓ **getstr(pstr, &var);**

SỰ LỒNG NHAU CỦA LỜI GỌI HÀM

```
main()
  {
    …
    palindrome();
    …
  }
```

```
palindrome()
    {
        …
        getstr();
        reverse();
        cmp();
        …
    }
```

CÁC HÀM TRONG CHƯƠNG TRÌNH CÓ NHIỀU TẬP TIN

- ✓ Các hàm cũng có thể được định nghĩa là **static** hoặc **external**
- ✓ Các hàm tĩnh (static) chỉ được nhận biết bên trong tập tin chương trình và phạm vi của nó không vượt ra khỏi tập tin chương trình
- ✓ **static fn _type fn_name (argument list);**
- ✓ Hàm ngoại (external) được nhận biết bởi tất cả các tập tin của chương trình
- ✓ **extern fn_type fn_name (argument list);**

CON TRỎ HÀM

- ✓ Lưu địa chỉ bắt đầu của hàm
- ✓ Hàm có một vị trí vật lý trong bộ nhớ, vị trí này có thể gán cho một con trỏ

```
1  #include <stdio.h>
2  #include <conio.h>
3
4  void test( int *n){
5
6     *n = *n +10;
7  }
8  int main(){
9        int a = 15;
10       test(&a);
11       printf("%d",a);
12       getch();
13 }
```

TÓM TẮT BÀI HỌC

- ✓ Hàm được dùng để thực thi một chuỗi các chỉ thị nhiều hơn một lần
  - ✓ Các đối số tới hàm có thể là các hằng, biến, biểu thức hay các hàm
  - ✓ **type_specifier** xác định kiểu dữ liệu của giá trị sẽ được trả về bởi hàm
  - ✓ Một hàm không thể được định nghĩa bên trong một hàm khác
  - ✓ Một nguyên mẫu hàm là một sự khai báo hàm để chỉ ra các kiểu dữ liệu của các đối số
  - ✓ Một con trỏ hàm có thể được dùng để gọi một hàm
- ✓ Trong C, mặc định, tất cả các đối số của hàm được truyền bằng giá trị
- ✓ Có ba loại biến cơ bản: biến cục bộ, tham số hình thức và biến toàn cục
- ✓ Lớp lưu trữ

BACHKHOA EDUCATION APTECH
Làm trước học sau

Aptech COMPUTER EDUCATION

TRƯỜNG ĐÀO TẠO LẬP TRÌNH VIÊN VÀ QUẢN TRỊ MẠNG QUỐC TẾ BACHKHOA-APTECH

THANK FOR WATCH!