**Bài 18: Ví dụ về cách viết và sử dụng** **chương trình con**

1. **Cấu trúc của hàm không có giá trị trả về:**

**void** <tên hàm>([<danh sách tham số>])

{

[<dãy các lệnh>]

}

Trong đó: Phần đầu gồm: + Void: là từ khóa.

+ Tên hàm: do người lập trình đặt

+ Tham số: có thể có hoặc không

Phần thân: là dãy lệnh bất kì đặt trong cặp ngoặc { }

Ví dụ: Viết chương trình vẽ 10 hình chữ nhật bằng hàm không có giá trị trả về.

#include <bits/stdc++.h>;

using namespace std;

void vehcn()

{

cout <<"\* \* \* \* \*\*"<<endl;

cout <<"\* \*"<<endl; ( chương trình con vẽ hình chữ nhật)

cout <<"\* \* \* \* \*\*"<<endl;

}

int main(){

for (int i= 0; i<10; i++) (vẽ 10 hình chữ nhật)

vehcn();

return 0;

}

**Bài tập: viết chương trình vẽ 20 hình tam giác.**

1. **Cấu trúc của hàm có giá trị trả về:**

**kiểu dữ liệu** <tên hàm>([<danh sách tham số>])

{

<khối lệnh>

**return** <giá trị trả về>;

}

Trong đó: Phần đầu gồm: + *Kiểu dữ liệu* là kiểu dữ liệu của giá trị mà hàm trả về và chỉ có thể là các kiểu *int*, *float*, *char*, *bool*, *string*.

+ Tên hàm: do người lập trình đặt

+ Tham số: có thể có hoặc không

Phần thân: là dãy lệnh bất kì và câu lệnh Return trả về giá trị nào trong bài toán.

**Ví dụ:**

Viết chương trình nhập chiều dài và chiều rộng hình chữ nhật, sau đó tính diện tích và in ra màn hình (sử dụng hàm có giá trị trả về)

#include <bits/stdc++.h>;

using namespace std;

int dienTich(int a, int b){

int s=a\*b; (Hàm tính diện tích)

return s;

}

int main(){

int dai, rong;

cout<<"Nhap chieu dai va chieu rong cua hcn";

cin>>dai>>rong;

cout<<“dien tich la:”<<dienTich(dai,rong);

return 0;

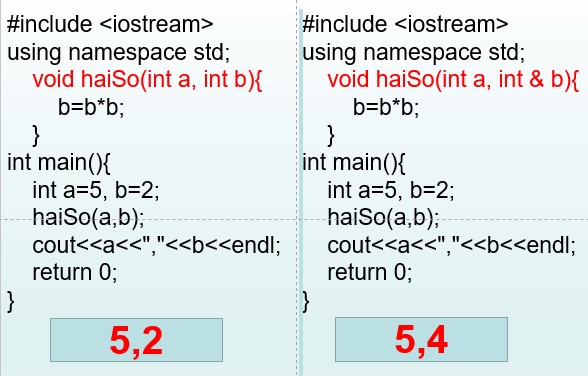
}

* **Phân biệt tham trị và tham biến:**

+ Tham trị: khi khai báo không có dấu & giữa kiểu và tham số. Khi thực hiện chương trình giá trị không thay đổi.

+ Tham biến: khi khai báo có dấu & giữa kiểu và tham số . Khi thực hiện chương trình giá trị thay đổi.

Ví dụ: so sánh 2 chương trình sau:

****

**Bài tập: viết chương trình tìm số lớn nhất trong 3 số a, b, c (có sử dụng chương trình con.**