C++ Function Parameters

Biến mã tĩnh lặng trở nên linh hoạt

Hành trình chinh phục C++

Ngày 14 tháng 1 năm 2026

Bức Tranh Toàn Cảnh (The Big Picture)

Ẩn dụ: Chiếc máy xay sinh tố

Hãy tưởng tượng một hàm (Function) giống như một **Chiếc máy xay sinh tố**.

- Nếu không có tham số: Máy bị "hàn chết", chỉ xay đúng một quả chuối có sẵn.
- **Function Parameters**: Chính là cái **phiễu (khe nạp)** phía trên.

Nhờ nó, bạn có thể thả dâu tây, xoài, táo vào. Cùng một chiếc máy, nhưng nguyên liệu khác nhau sẽ tạo ra kết quả khác nhau.

Lộ Trình Khám Phá (Roadmap)

Chúng ta sẽ đi qua 6 chặng để nắm vững từng viên gạch:

1. **Parameters và Arguments:** Phân biệt "Cái Phiếu"và "Trái Cây".
2. **Default Parameters:** Cài đặt mặc định (Khi lười chọn món).
3. **Multiple Parameters:** Công thức pha chế hỗn hợp.
4. **Return Values:** Shipper giao hàng tận tay.
5. **Pass by Reference:** Sửa trực tiếp vào bản gốc (Quan trọng).
6. **Passing Arrays:** Xử lý cả một thùng hàng.

Chương 1: Parameters và Arguments

Tư duy cốt lõi:

- **Parameter (Tham số):** Biến được định nghĩa lúc **thiết kế** hàm (Cái nhãn "Bỏ trái cây vào đây").
- **Argument (Đối số):** Dữ liệu thực tế truyền vào lúc **sử dụng** hàm (Quả táo thật).

Syntax: Parameters và Arguments

```cpp
#include <iostream>
#include <string>
using namespace std;

// CACH THIET KE:
// "tenNguoi" o day chinh la PARAMETER (Cai khuon)
void chaoHoi(string tenNguoi) {
  cout << "Xin chao " << tenNguoi << "!\n";
}

int main() {
  // CACH SU DUNG:
  // "Nam" va "Lan" o day chinh la ARGUMENT (Du lieu that)
  chaoHoi("Nam");
  chaoHoi("Lan");

  return 0;
}
```

Kiểm tra sự hiểu biết (Active Recall)

Cho đoạn code sau:

```
1 void tinhTong(int soA) {
2   cout << soA + 10;
3 }
4 int main() {
5   int x = 5;
6   tinhTong(x);
7   return 0;
8 }
```

Trong dòng `tinhTong(x);`:

- `soA` được gọi là gì?
- `x` được gọi là gì?

Kiểm tra sự hiểu biết (Active Recall)

Cho đoạn code sau:

```
1 void tinhTong(int soA) {
2   cout << soA + 10;
3 }
4 int main() {
5   int x = 5;
6   tinhTong(x);
7   return 0;
8 }
```

Trong dòng `tinhTong(x);`:

- `soA` được gọi là gì?
- `x` được gọi là gì?

Đáp án

soA là Parameter (Cái khuôn).
x là Argument (Nguyên liệu thực tế).

Chương 2: Default Parameters (Tham số mặc định)

Tình huống quán cà phê

- Khách gọi "Cho ly cà phê": Nhân viên làm theo **công thức chuẩn** (Mặc định).
- Khách gọi "Cà phê 100% đường": Nhân viên **ghi đè** công thức chuẩn.

Trong C++, **Default Parameter** dùng giá trị có sẵn nếu người dùng không truyền tham số.

Syntax: Default Parameters

Bí mật nằm ở dấu bằng = khi khai báo hàm.

```cpp
// Gan gia tri mac dinh "Viet Nam" ngay tai day
void xuatXu(string quocGia = "Viet Nam") {
  cout << "Toi den tu " << quocGia << "\n";
}

int main() {
  // Truong hop 1: CO truyen tham so -> Ghi de
  xuatXu("Nhat Ban"); // In ra: Toi den tu Nhat Ban

  // Truong hop 2: KHONG truyen tham so -> Dung mac dinh
  xuatXu();           // In ra: Toi den tu Viet Nam
  return 0;
}
```

Kiểm tra sự hiểu biết

Đoạn code sau in ra màn hình những gì?

```
1 void thoiTiet(int nhietDo = 25) {
2   cout << "Nhiet do la: " << nhietDo << " do C\n";
3 }
4 int main() {
5   thoiTiet(30);
6   thoiTiet();
7 }
```

Kiểm tra sự hiểu biết

Đoạn code sau in ra màn hình những gì?

```
1 void thoiTiet(int nhietDo = 25) {
2   cout << "Nhiet do la: " << nhietDo << " do C\n";
3 }
4 int main() {
5   thoiTiet(30);
6   thoiTiet();
7 }
```

Đáp án

1. Nhiet do la: 30 do C
2. Nhiet do la: 25 do C

Chương 3: Multiple Parameters (Nhiều tham số)

Để pha chế phức tạp, hàm cần nhiều nguyên liệu: Loại trái cây, Lượng đường, Đá...

Quy tắc vàng

- Các tham số ngăn cách bởi dấu phẩy ,.
- **Thứ tự là mệnh lệnh!** Phải truyền đúng thứ tự định nghĩa.

Syntax: Multiple Parameters

```cpp
// Ham can 2 nguyen lieu theo thu tu: Chu truoc, So sau
void thongTinNhanVien(string ten, int tuoi) {
  cout << ten << " nam nay " << tuoi << " tuoi.\n";
}

int main() {
  //    NG   :
  thongTinNhanVien("Tung", 25);

  // SAI ( L i  ngay  l p   t c ):
  // thongTinNhanVien(25, "Tung");

  return 0;
}
```

Chương 4: Return Values (Giá trị trả về)

Hàm Void (Hư vô)

Giống như hét lên một câu rồi thôi. Không thể "cầm nắm"kết quả để dùng tiếp.

Hàm Return (Shipper)

Giống như thợ làm bánh. Họ làm xong và **trả lại (return)** cái bánh cho bạn để bạn mang đi đâu tùy ý.

Syntax: Return Values

Thay `void` bằng kiểu dữ liệu (`int`, `string`...) và dùng lệnh `return`.

```
int tinhTong(int x, int y) {
  return x + y; // Tra ve ket qua, khong in ra
}

int main() {
  // Cat ket qua vao bien
  int ketQua = tinhTong(5, 3);
  cout << "Tong: " << ketQua;

  // Hoac dung truc tiep trong phep tinh khac
  cout << "Tong + 10: " << (tinhTong(5, 3) + 10);
  return 0;
}
```

Kiểm tra sự hiểu biết

Điền vào chỗ trống để tính tổng diện tích 2 hình vuông:

```
int dienTichVuong(int canh) { return canh * canh; }

int main() {
  int hinhA = dienTichVuong(5); // 25
  int hinhB = dienTichVuong(3); // 9

  int tongDienTich = ...?
  cout << tongDienTich;
}
```

Kiểm tra sự hiểu biết

Điền vào chỗ trống để tính tổng diện tích 2 hình vuông:

```
1 int dienTichVuong(int canh) { return canh * canh; }
2
3 int main() {
4   int hinhA = dienTichVuong(5); // 25
5   int hinhB = dienTichVuong(3); // 9
6
7   int tongDienTich = ...?
8   cout << tongDienTich;
9 }
```

Đáp án

Code: `int tongDienTich = hinhA + hinhB;`

Kết quả in ra: **34**

Chương 5: Pass by Reference (Truyền tham chiếu)

Đây là phần quan trọng nhất!

- **Pass by Value (Mặc định):** Gửi file đính kèm email. Bạn sửa bản copy, bản gốc máy tôi vẫn y nguyên.
- **Pass by Reference (Tham chiếu):** Gửi **link Google Docs**. Bạn sửa trên link, bản gốc của tôi thay đổi ngay lập tức.

Dấu hiệu nhận biết: Ký tự & khi khai báo tham số.

Syntax: Pass by Reference (&)

Ví dụ kinh điển: Hoán đổi (Swap).

```
1  // int &x: "Toi muon lay duong link toi bien x"
2  void hoanDoi(int &x, int &y) {
3    int tam = x;
4    x = y;
5    y = tam;
6  }
7
8  int main() {
9    int a = 10, b = 20;
10   hoanDoi(a, b);
11   cout << a << " - " << b; // In ra: 20 - 10
12 }
```

Nếu không có dấu &, a và b vẫn sẽ là 10 - 20.

Cạm bẫy (Kiểm tra sự hiểu biết)

Hãy cẩn thận với đoạn code sau!

```
1  void tangGiaTri(int &a, int b) { // Chi a co dau &, b thi khong
2    a = a + 1;
3    b = b + 1;
4  }
5
6  int main() {
7    int x = 5;
8    int y = 5;
9    tangGiaTri(x, y);
10   cout << "x=" << x << ", y=" << y;
11 }
```

Cạm bẫy (Kiểm tra sự hiểu biết)

Hãy cẩn thận với đoạn code sau!

```
1  void tangGiaTri(int &a, int b) { // Chi a co dau &, b thi khong
2    a = a + 1;
3    b = b + 1;
4  }
5
6  int main() {
7    int x = 5;
8    int y = 5;
9    tangGiaTri(x, y);
10   cout << "x=" << x << ", y=" << y;
11 }
```

Đáp án

x = 6 (Do có & nên bản gốc bị sửa).
y = 5 (Do không có &, chỉ là bản copy bị sửa, bản gốc y nguyên).

Chương 6: Passing Arrays (Truyền mảng)

Đặc biệt

Mảng (Array) **MẶC ĐỊNH** luôn luôn là tham chiếu (Google Docs). Không cần dấu &.

Khi đưa một mảng vào hàm, mọi thay đổi bên trong hàm đều tác động trực tiếp lên mảng gốc.

Syntax: Passing Arrays

```cpp
// mangSo[]: Nhan vao mot  m ng
void inMang(int mangSo[], int kichThuoc) {
  for (int i = 0; i < kichThuoc; i++) {
    // Thu sua doi gia tri truc tiep
    mangSo[i] = mangSo[i] * 2;
    cout << mangSo[i] << " ";
  }
}

int main() {
  int soYeuThich[3] = {1, 2, 3};
  inMang(soYeuThich, 3); // In ra: 2 4 6

  // Mang goc da bi thay doi vinh vien!
  cout << "\nSo dau tien: " << soYeuThich[0]; // In ra: 2
}
```

Tổng kết hành trình

1. **Parameters vs Arguments:** Cái khuôn và nguyên liệu.
2. **Default Parameters:** Chế độ cho người "lười".
3. **Multiple Parameters:** Đúng thứ tự là chân ái.
4. **Return Values:** Shipper giao hàng.
5. **Pass by Reference (&):** Sửa bản gốc (Google Docs).
6. **Passing Arrays:** Luôn sửa bản gốc (Mặc định).

Thử thách Final Boss

Đề bài: Viết hàm `doiChoVaNhanDoi`:

- Nhận vào 2 số nguyên.
- Hoán đổi vị trí của chúng.
- SAU ĐÓ nhân đôi giá trị cả hai.
- Ví dụ: Vào `a=1, b=2` $\rightarrow$ Ra `a=4, b=2`.

Thử thách Final Boss

Đề bài: Viết hàm `doiChoVaNhanDoi`:

- Nhận vào 2 số nguyên.
- Hoán đổi vị trí của chúng.
- SAU ĐÓ nhân đôi giá trị cả hai.
- Ví dụ: Vào a=1, b=2 $\rightarrow$ Ra a=4, b=2.

Lời giải (Model Code)

```
void doiChoVaNhanDoi(int &a, int &b) {
  // 1. Hoan doi (Swap)
  int temp = a;
  a = b;
  b = temp;

  // 2. Nhan doi (Double)
  a *= 2; // a = a * 2
  b *= 2; // b = b * 2
}
```

Bước tiếp theo

Bạn đã hoàn thành xuất sắc chủ đề Function Parameters!

Next Step

Chủ đề tiếp theo: **Function Overloading** (Nạp chồng hàm) - Nghệ thuật dùng 1 cái tên cho nhiều hàm khác nhau.

Chúc bạn học tốt!