

HƯỚNG DẪN THỰC HÀNH

TEMPLATE C CHO HÀM

1 Dẫn nhập

Giả sử yêu cầu viết hàm trả về số lớn hơn trong hai số a và b.

- Với a, b là số nguyên, ta có:

```
int TimMax(int a, int b)
{
    if(a > b) return a;
    return b;
}
```

- Với a, b là số thực, ta có:

```
double TimMax(double a, double b)
{
    if(a > b) return a;
    return b;
}
```

Hai hàm nói trên có cùng mã hàm nhưng vì khác kiểu nên viết 2 hàm tách biệt

→ Làm sao để viết một hàm mà vẫn có thể gọi cho các kiểu dữ liệu khác nhau.

→ Áp dụng template cho hàm

2 Template cho hàm

Template hỗ trợ

- Không cần xác định cụ thể kiểu dữ liệu của tham số truyền vào hoặc kết quả trả về khi định nghĩa hàm
 - Chỉ cần xác định kiểu dữ liệu khi gọi hàm
- Tránh trùng lặp mã nguồn, dễ bảo trì

Cú pháp

```
template <typename Type1, typename Type2, ...>
Type2 Ten_ham (Type1 thamso1, Type1 thamso2)
{
    // Mã nguồn của hàm
}
```

Trong đó,

- *template, typename* là các từ khóa
- *Type1, Type2* là tên kiểu dữ liệu trừu tượng. Sinh viên đặt tên bất kì.

Ví dụ 1: viết lại hàm tìm số lớn nhất trong phần “Dẫn nhập”

```
template <typename T>
T TimMax(T a, T b)
{
    if(a > b) return a;
    return b;
}

int _tmain(int argc, _TCHAR* argv[])
{
    // Goi ham cho so nguyen
    int a = 5;
    int b = 6;
    printf("Gia tri lon hon la: %d\n", TimMax(a, b));

    // Goi ham cho so thuc
    double c = 5.0;
    double d = 7.0;
    printf("Gia tri lon hon la: %lf\n", TimMax(c, d));

    return 0;
}
```

Lưu ý: Khi sinh viên tách hàm con ra tập tin riêng mà có sử dụng template thì cần đặt tất cả nội dung của hàm trong tập tin “.h”, không đặt phần định nghĩa hàm trong tập tin .cpp

3 Bài tập

1. Viết chương trình sắp xếp dãy số theo thứ tự tăng dần.
2. Viết chương trình sắp xếp mảng các phần tử theo thứ tự tăng dần. Các phần tử trong mảng có cùng kiểu, và có thể là số nguyên, số thực, phân số, học sinh, ...
3. Viết chương trình tính tổng các phần tử trong một mảng. Biết rằng các phần tử trong mảng có cùng kiểu, và có thể là số nguyên, số thực, phân số.