

# TIN ĐẠI CƯƠNG

---

## BÀI 4: HÀM VÀ LỆNH LẶP

# Nội dung buổi trước

---

- Một số kiểu dữ liệu thường dùng trong chương trình C++: số nguyên (**int**), số thực (**double**, **float**), logic (**bool**)
- Kiểu dữ liệu quyết định cách máy tính thực hiện các phép toán và xử lý các biến
- In dữ liệu ra màn hình thông qua biến **cout** và phép toán xuất dữ liệu (**<<**)
- Nhập dữ liệu từ bàn phím thông qua biến **cin** và phép toán đọc dữ liệu (**>>**)
- Hàm: khái niệm, cách viết và cách gọi hàm

# Nội dung chính

---

1. Tại sao cần viết chương trình con?
2. Vòng lặp
3. Ví dụ về vòng lặp dùng biến đếm
4. Ví dụ về vòng lặp dùng điều kiện
5. Biểu thức logic
6. Bài tập

Phần 1

# Tại sao cần viết chương trình con?

# Chương trình đơn giản

---

Yêu cầu: nhập số  $n$  và tính  $\sqrt[n]{n}$ , không dùng hàm có sẵn

```
#include <iostream>           // khai báo thư viện

using namespace std;         // khai báo tên miền chuẩn

int main() {                  // bắt đầu hàm chính
    double n;                 // biến để chứa số n
    cout << "N = ";          // in ra chuỗi "N = "
    cin >> n;                 // nhập số và ghi vào n

    double x = 1;             // biến x (để chứa căn 2 của n)
```

# Chương trình đơn giản

---

```
x = (x + n/x) / 2;    // tính x
x = (x + n/x) / 2;    // tính x
x = (x + n/x) / 2;    // tính x
x = (x + n/x) / 2;    // tính x
x = (x + n/x) / 2;    // tính x
x = (x + n/x) / 2;    // tính x
x = (x + n/x) / 2;    // tính x
x = (x + n/x) / 2;    // tính x
x = (x + n/x) / 2;    // tính x

cout << "SQRT(n) = " << x;    // in số x ra màn hình
}
```

# Hạn chế của cách viết đơn giản

---

- **Sự khó hiểu**: chương trình gồm nhiều phần, mỗi phần có mục đích khác nhau, ta phải đọc kỹ phần ghi chú mới nắm được nội dung
  - Chẳng hạn như phần tính căn bậc 2 của  $n$ , nếu không có ghi chú thì khó có thể biết nó làm gì
- **Sự cứng nhắc**: chỉ tính được căn bậc 2 của biến  $n$ , nếu muốn tính căn bậc 2 của biến  $m$  thì phải viết lại từ đầu
  - Hai đoạn mã hầu như giống nhau, khác tên biến
  - Nếu muốn tính căn bậc 2 cho 100 biến thì sao?
  - Nếu lỡ viết sai sẽ phải sửa ở 100 chỗ giống nhau?

# Tách thành các hàm

---

Yêu cầu: nhập số  $n$  và tính  $\sqrt[2]{n}$ , không dùng hàm có sẵn

```
#include <iostream>           // khai báo thư viện
using namespace std;         // khai báo tên miền chuẩn

double can2(double n) {      // tự định nghĩa hàm sqrt của ta
    double x = 1;            // biến x (để chứa căn 2 của n)
    x = (x + n/x) / 2;       // tính x
    ...
    x = (x + n/x) / 2;       // tính x
    return x;                // trả về kết quả tính được
}
```



# Tách thành các hàm

---

```
int main() { // bắt đầu hàm chính
    double n; // biến để chứa số n
    cout << "N = "; // in ra chuỗi "N = "
    cin >> n; // nhập số và ghi vào n

    // gọi hàm tính toán và in kết quả ra màn hình
    cout << "SQRT(n) = " << can2(n);
}
```

- Tên hàm tự nó cũng cung cấp thông tin về đoạn mã
- Không còn phụ thuộc vào tên biến, ta có thể gọi hàm can2 với bất kì biến nào mà ta cần
- Sửa sai ở một đoạn mã duy nhất

Phần 2

# Vòng lặp