

TIN HỌC ĐẠI CƯƠNG

Bài 2: Chương trình C++ đơn giản

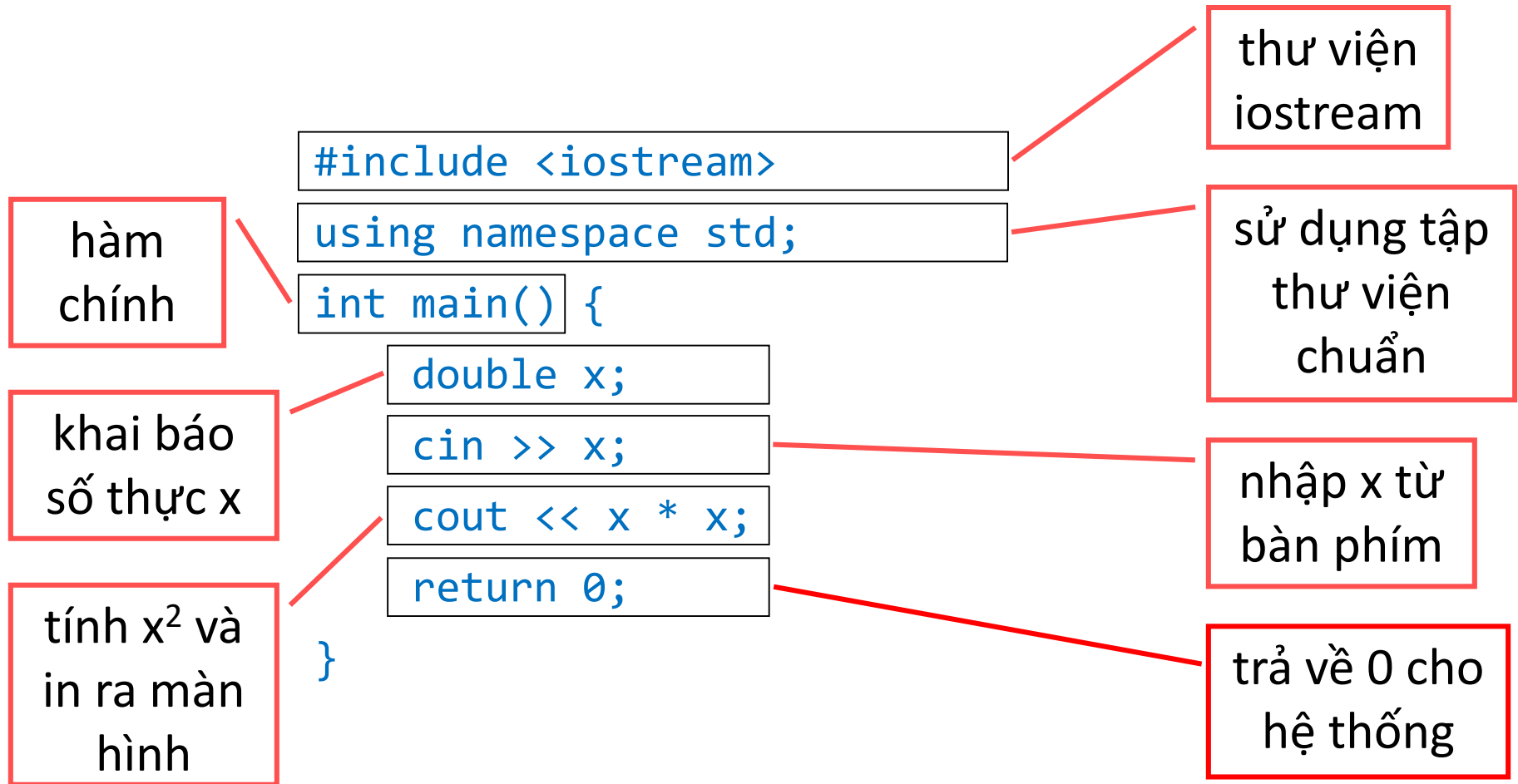
Nội dung chính

1. Chương trình C++ đầu tiên
2. Biến & Định danh
3. Biểu thức, Phép toán & Phép gán
4. Các cấu trúc điều khiển
5. Thuật toán
6. Bài tập

Phần 1

Chương trình C++ đầu tiên

Chương trình: nhập x và in ra x^2



Chương trình: nhập N, tính và in ra $\sqrt[2]{N}$

```
#include <iostream> // sử dụng thư viện vào/ra chuẩn
#include <cmath> // sử dụng thư viện các hàm toán
using namespace std; // sử dụng không gian tên chuẩn

int main() { // bắt đầu hàm main()
    double n, x; // khai báo 2 biến n và x
    cout << "N = "; // in ra màn hình "N = "
    cin >> n; // nhập dữ liệu vào biến n
    x = sqrt(n); // tính giá trị của x
    cout << "X = " << x; // in ra màn hình: "X = ..."
}
```

Phần 2

Biến & Định danh

Biến và định danh

Quy tắc khai báo chung:

<kiểu> <tên biến>;

<kiểu> <tên biến> = <giá trị>;

Ví dụ:

```
int x;           // số nguyên x
int n = 100;    // số nguyên n, giá trị 100
double d = 1.5; // số thực d, giá trị 1,5
double m;      // số thực m
bool kiểmtra;  // biến logic kiểmtra
bool ok = false; // biến ok, giá trị = sai
int a, b, c;   // 3 số nguyên a, b, c
```

Biến và định danh

- **Khái niệm:** vùng trong bộ nhớ máy tính dùng để chứa những kết quả tính toán
 - Cần được đặt tên để dễ thao tác
 - Gọi là “biến” (variable)
- **Nguyên tắc:**
 - Phải khai báo trước khi dùng
 - Phải chỉ ra kiểu (loại số)
 - Viết liền, phân biệt chữ hoa và chữ thường
 - Bắt đầu bởi chữ cái (tiếng Anh) theo sau là chữ cái hoặc chữ số hoặc dấu gạch dưới (_): **n**, **soA**, **thamso1**, **diepvien007**,...

Biến và định danh

- Không chỉ cho phép đặt tên biến, C++ còn cho phép đặt tên cho các thành phần khác:
 - Tên hàm (function)
 - Tên lớp (class)
 - Tên cấu trúc (struct)
- Định danh (identifier) là quy cách chung cho việc đặt tên của các thành phần trên
 - Có thể sử dụng thêm dấu gạch dưới (_)
 - Không được trùng với từ khóa của ngôn ngữ
 - Không được trùng nhau

Phần 3

Biểu thức, Phép toán & Phép gán

Biểu thức (expression)

- **Khái niệm:** sự kết hợp giữa các giá trị, biến, phép toán và các cặp ngoặc để có thể thực hiện tính toán được kết quả cụ thể nào đó
 - Nhận xét: tương tự như biểu thức trong toán học

- **Ví dụ:**

$$10 - 20 * 3$$

$$n + 2 * m$$

$$m * -1 / (k + 1.5)$$

$$(dayNho + dayLon) * chieuCao / 2$$

$$(a + b + c) / 3$$

Phép toán (operator)

- **Phép toán:** C++ cho phép sử dụng hầu hết các phép toán thông dụng
 - Phép tính số học: cộng (+), trừ (-), nhân (*), chia lấy thương (/), chia lấy dư (%)
 - Phép so sánh, phép toán logic,...
- **Thứ tự thực hiện phép toán:** tương tự như trong biểu thức toán học
- **Thư viện <cmath>:** cung cấp các hàm toán học thường dùng, chẳng hạn như hàm căn 2 (sqrt), hàm mũ (pow),...

(sẽ giới thiệu đầy đủ hơn ở phần bàn về kiểu dữ liệu)

Phép gán

- **Định nghĩa:** phép toán ghi nhớ giá trị biểu thức vào một biến
- **Cú pháp:**
 $\langle \text{biến} \rangle = \langle \text{biểu thức} \rangle;$
- **Ví dụ:**
 $n = 10;$
 $m = n + 5 / 3;$
 $t = a + b + c;$
- **Chú ý:** khi viết mã, rất dễ nhầm phép gán (=) và phép so sánh bằng (==)

Phần 4

Các cấu trúc điều khiển

Các cấu trúc điều khiển

- Có 3 cấu trúc điều khiển cơ bản
 - **Tuần tự**: Thực hiện tuyến tính từng việc một
 - **Lặp**: Thực hiện lặp lại một hoặc nhiều việc cho đến khi điều kiện nhất định được thỏa mãn
 - **Lựa chọn (rẽ nhánh)**: Chọn thực hiện một hoặc nhiều việc dựa trên một điều kiện nhất định
- Các cấu trúc này tương tự như nhiều hành vi trong cuộc sống
- Đọc trước các chương 2, 3 và 4 của giáo trình để tìm hiểu về các điều khiển này

Các cấu trúc điều khiển trong cuộc sống

- Tuần tự thực hiện từng bước từ đầu đến cuối danh sách
 - Chấm bài
 - Đếm số người trong phòng
 - Kiểm tra an ninh
- Lặp lại một công việc đến khi đạt được mục đích đề ra
 - Học thuộc lòng
 - Tập thể thao
- Rẽ nhánh lựa chọn thực hiện một trong số nhiều việc
 - Chọn đi học hay đi chơi
 - Chọn mua món đồ nào
- Đa phần công việc trong cuộc sống là sự kết hợp phức tạp của 3 loại cấu trúc điều khiển trên, phần mềm cũng vậy

Phần 5

Thuật toán

Định nghĩa và đặc trưng

- **Định nghĩa:** các bước cần tiến hành để giải quyết một công việc cụ thể nào đó
- Thuật toán phổ biến trong cuộc sống, có trước máy tính và có nhiều dạng khác nhau
- Đặc trưng (*xem phần 1.1 của giáo trình*):
 - Tính hữu hạn
 - Tính máy móc
 - Tính dừng
 - Mở rộng: Tính đúng
 - Mở rộng: Tính tổng quát

Ví dụ đơn giản

Ví dụ: Tính bình phương của số m

Bước 1: Nhập giá trị cho m

Bước 2: Tính giá trị $m \times m$ và đưa vào s

Bước 3: Trả về giá trị s cho chương trình gọi

Ta thấy:

- Thuật toán có 3 bước
- Viết rõ ràng, không thể hiểu sai
- Làm theo thuật toán ta có thể tính được kết quả mà không cần hiểu khái niệm “bình phương” là gì

Ví dụ phức tạp hơn

Giải phương trình $ax^2 + bx + c = 0$ (với $a \neq 0$)

- Bước 1: Nhập các giá trị a, b, c
- Bước 2: Nếu $a = 0$ thì thông báo lỗi và dừng
- Bước 3: Tính $d = b^2 - 4 \times a \times c$
- Bước 4: Nếu $d < 0$ thì thông báo vô nghiệm và dừng
- Bước 5: Nếu $d > 0$ thì thực hiện Bước 7
- Bước 6: Thông báo có nghiệm $x = -b/2/a$ và dừng
- Bước 7: Thông báo có hai nghiệm
 - $x_1 = \frac{-b + \sqrt{d}}{2a}$
 - $x_2 = \frac{-b - \sqrt{d}}{2a}$

Ví dụ vui

Cho con sư tử vào tủ lạnh

Bước 1: Mở cửa tủ lạnh

Bước 2: Cho con sư tử vào

Bước 3: Đóng cửa tủ lạnh

Câu hỏi: *hãy nêu một vài thuật toán tương tự trong cuộc sống (thuật toán nhưng không dính dáng gì tới máy tính)*

Phần 6

Bài tập

Bài tập

1. Hãy xây dựng thuật toán (hoặc viết chương trình) giải phương trình bậc nhất $P(x): a x + b = c$
2. Hãy xây dựng thuật toán để biến đổi số N ở dạng thập phân thành số M tương đương ở dạng nhị phân
 - Ví dụ: $N = 14$ thì thuật toán trả về $M = 1110$
3. Hãy xây dựng thuật toán để tính tổng các chữ số của một số nguyên N bất kỳ
 - Ví dụ: $N = 2016$ thì thuật toán trả về 9 ($2+0+1+6$)
4. Làm thế nào để tạo các số ngẫu nhiên trong máy tính? Hãy xây dựng một thuật toán để tạo ra được các số như vậy
5. Nhập điểm toán, lý và hóa của một học sinh. In ra điểm trung bình 3 môn học của học sinh đó.

Ví dụ: nhập 2 số a và b, tính tổng 2 số

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int a, b;
6
7      cout << "A = "; cin >> a;    // nhập số A
8      cout << "B = "; cin >> b;    // nhập số B
9
10     cout << "Tong = " << a+b;    // tính tổng và in ra
11
12     return 0;
13 }
```


Ví dụ: tính diện tích tam giác có 3 cạnh a, b, c

```
1  #include <iostream>
2  using namespace std;
3
4  double DienTich(double a, double b, double c) {
5      double p = (a + b + c) / 2;
6      return sqrt(p * (p-a) * (p-b) * (p-c));
7  }
8
9  int main() {
10     double a, b, c;
11
12     cout << "A = "; cin >> a;    // nhập số A
13     cout << "B = "; cin >> b;    // nhập số B
14     cout << "C = "; cin >> c;    // nhập số C
15
16     cout << "DT = " << DienTich(a, b, c);    // gọi hàm tính diện tích
17
18     return 0;
19 }
```

Ví dụ: tính khoảng cách giữa (x_1, y_1) và (x_2, y_2)

```
1  #include <iostream>
2  using namespace std;
3
4  double KhoangCach(double x1, double y1, double x2, double y2) {
5      return sqrt((x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2));
6  }
7
8  int main() {
9      double x1, y1, x2, y2;
10
11     cout << "X1 = "; cin >> x1;      // nhập x1
12     cout << "Y1 = "; cin >> y1;      // nhập y1
13     cout << "X2 = "; cin >> x2;      // nhập x2
14     cout << "Y2 = "; cin >> y2;      // nhập y2
15
16     cout << "KC = " << KhoangCach(x1, y1, x2, y2);
17
18     return 0;
19 }
```