TIN ĐẠI CƯƠNG

BÀI 2: CÁC KHÁI NIỆM CƠ BẢN

Nhắc lại nội dung bài trước

- Khái niệm "Thuật toán" và các đặc trưng:
  - Tính hữu hạn
  - Tính máy móc
  - Tính dừng
  - Có "giao diện": đầu vào (input) & đầu ra (output)
- Tham trị & tham chiếu
- Có 3 loại cấu trúc điều khiển cơ bản: tuần tự, lặp, rẽ nhánh
- Sử dụng phần mềm Dev-C++: cách viết, dịch, chạy và sửa lỗi chương trình C++

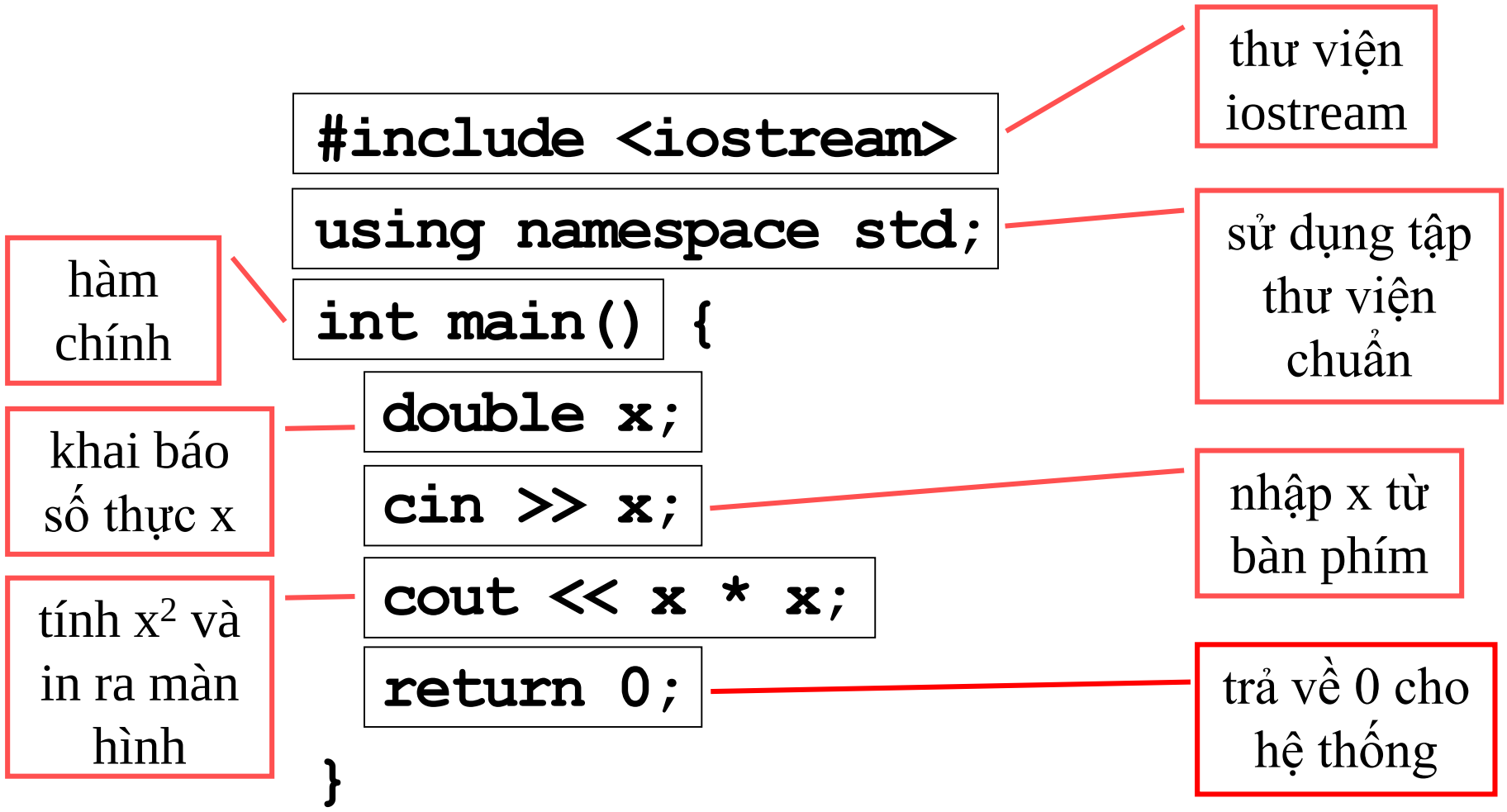
Nội dung chính

1. Các khái niệm cơ sở
    1. Biến (variable) và định danh (identifier)
    2. Biểu thức (expression)
    3. Phép gán
    4. Vài kiểu dữ liệu cơ bản: nguyên, thực, logic
2. Nhập và xuất dữ liệu
3. Phân rã bài toán (vấn đề)
4. Hàm (function)
5. Ví dụ và bài tập

Phần 1

Các khái niệm cơ sở

Các khái niệm cơ sở



1.1 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
  - Biến (variable) hay định danh (identifier)

- Nguyên tắc:
  - Phải khai báo trước khi dùng
  - Phải chỉ ra kiểu (loại số)
  - Tên bắt đầu bởi chữ cái theo sau là chữ cái hoặc số, viết liền: n, soA, thamso1, diepvien007,…

1.1 Biến và định danh

- Quy tắc khai báo chung:

  <kiểu> <định danh>;

  <kiểu> <định danh> = <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 kiemtra;       // biến logic kiem tra
bool ok = false;    // biến logic ok, giá trị = sai
int a, b, c;        // 3 số nguyên a, b, c
```

1.2 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
```

1.3 Phép gán

- Định nghĩa: phép toán ghi nhớ giá trị biểu thức vào một biến
- Cú pháp:

```
<biến> = <biểu thức>;
```

- Thực hiện: tính toán giá trị của biểu thức ở vế phải, lấy kết quả ghi vào biến ở vế trái
- Ví dụ:

```
n = 10;
m = n + 5 / 3;
t = a + b + c;
```

1.4 Vài kiểu dữ liệu cơ bản

- Hỏi: tại sao cần định nghĩa "kiểu dữ liệu"?
- Đáp: để thực hiện tính toán cho chính xác
- Một số kiểu cơ bản:
  - Số nguyên: int
  - Số thực: double, float
  - Logic: bool
- Các phép toán thông dụng:
  - Số: cộng (+), trừ (-), nhân (*), chia (/), dư (%),...
  - Luân lý: và (&&), hoặc (||), phủ định (!),...
  - So sánh: bằng (==), khác (!=), lớn hơn (>),...

Kiểu nguyên (int)

- Dùng để lưu trữ số nguyên không quá lớn (trong khoảng từ ~ âm 2 tỉ đến dương 2 tỉ)
- Các phép tính cơ bản:
  - Các phép toán số học: cộng (+), trừ (-), nhân (*), chia lấy thương (/), lấy số dư (%)
  - Các phép toán đặc biệt: tăng 1 đơn vị (++), giảm 1 đơn vị (--)
  - Các phép so sánh giá trị: bằng (==), khác (!=), lớn hơn (>), nhỏ hơn (<), lớn hơn hoặc bằng (>=), nhỏ hơn hoặc bằng (<=)

Kiểu thực (float, double)

- Dùng để lưu trữ các số thực
  - Kiểu double có độ chính xác cao hơn kiểu float nhưng tốn nhiều bộ nhớ hơn
- Các phép tính cơ bản:
  - Các phép toán số học: cộng (+), trừ (-), nhân (*), chia (/)
  - Các phép so sánh giá trị.
  - Nhiều hàm toán học bổ sung (khai báo thư viện <cmath>): fabs, sqrt, pow, floor/ceil, exp, log, log10,…

Kiểu logic (bool)

- Lưu trữ các giá trị đúng/sai (true/false)
- Sử dụng trong các tình huống luân lý:
  - Là kết quả của các phép so sánh: >, >=, <, <=, ==, !=
  - Các phép toán logic: và (&&), hoặc (||), đảo (!), xor (^)
  - Sử dụng khi ra quyết định (sẽ học trong bài 4)

Phần 2

Nhập và xuất dữ liệu

Xuất dữ liệu

- Xuất dữ liệu thông qua biến cout (thường sẽ ghi ra màn hình)
  - Sau khi in xong, con trỏ sẽ dừng lại ở ngay sau phần vừa in để tiếp tục chờ lệnh in mới
  - Việc trình bày ra màn hình đôi khi khá quan trọng
- Ví dụ:

```
cout << "hello!!!";   // in chuỗi hello!!!
cout << abc;          // in ra giá trị của abc
cout << 5+6;          // tính giá trị 5+6 và in ra
cout << "A = " << a;  // in “A = ”, sau đó in giá trị a
cout << endl;         // chuyển con trỏ xuống dòng mới
```

Nhập dữ liệu

- Nhập dữ liệu thông qua biến cin (thường sẽ nhập từ bàn phím)
  - Máy tính đợi người dùng bấm <enter> sau đó sẽ phân tích xem người dùng nhập gì
  - Ngoại trừ một số tình huống đặc biệt, nhập dữ liệu luôn ghi vào một biến nào đó, nếu người dùng nhập không đúng định dạng số liệu có thể gây lỗi
  - Rất cẩn thận khi nhập nhiều dữ liệu cùng một lúc
- Ví dụ:

```
cin >> a;        // nhập dữ liệu vào biến a
cin >> a >> b;   // nhập dữ liệu vào a và tiếp vào b
```

Phần 3

Phân rã bài toán

2.2 Phân rã bài toán (vấn đề)

- Ý tưởng:
  - Một bài toán lớn có thể phân rã thành các bài toán nhỏ hơn (các thuật toán con)
  - Việc giải bài toán lớn = phối hợp lời giải các bài toán con với nhau
- Ví dụ: tính diện tích một đa giác lồi
  1. Chia đa giác thành các tam giác con
  2. Tính diện tích từng tam giác con
  3. Lấy tổng diện tích các tam giác con

2.2 Phân rã bài toán (vấn đề)

- Giải phương trình bậc 2:
  - Chia delta thành 3 trường hợp (âm, 0, dương)
  - Giải riêng rẽ từng trường hợp một
- Hầu hết các bài toán phức tạp đều được chia thành các chức năng con
  - Hệ thống menu của các phần mềm là một ví dụ điển hình của việc chia phần mềm thành các chức năng con
- Những bài toán không phân rã được thường là những bài rất khó

Phần 4

Hàm (function)

2.3 Hàm (function)

- Hàm: đoạn chương trình máy tính thực thi một thuật toán nào đó (và trả về kết quả)
- Cú pháp:

```
<kiểu kết quả> <tên hàm> (<tham số>) {
    // nội dung thực hiện thuật toán
}
```

- Ví dụ:

```
int dientich(int dai, int rong) {
    return dai * rong;
}
```

2.3 Hàm (function)

- Gọi hàm: gọi thông qua tên và tham số
- Ví dụ:

```
int n = dientich(30,40);
```

- Viết thành hàm thì có lợi gì?
  - Ý tưởng phân ra bài toán thành các bài toán con
  - Viết một lần, gọi mọi nơi
  - Nếu có sai thì chỉ cần sửa ở một chỗ
  - Có thể dùng trong các bài khác
- (nhắc lại) Khái niệm: tham chiếu & tham trị

Phần 5

Ví dụ và bài tập

Luyện tập qua các 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  }
```

Luyện tập qua các 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  }
```

Luyện tập qua các ví dụ

Tính khoảng cách giữa 2 điểm (x1,y1) và (x2,y2)

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

double KhoangCach(double x1, double y1, double x2, double y2) {
    return sqrt((x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2));
}

int main() {
    double x1, y1, x2, y2;

    cout << "X1 = "; cin >> x1;      // nhập x1
    cout << "Y1 = "; cin >> y1;      // nhập y1
    cout << "X2 = "; cin >> x2;      // nhập x2
    cout << "Y2 = "; cin >> y2;      // nhập y2

    cout << "KC = " << KhoangCach(x1, y1, x2, y2);

    return 0;
}
```

Bài tập

1. Nhập điểm toán, lý và hóa của một học sinh. In ra điểm trung bình 3 môn học của học sinh đó.
2. Nhập số nguyên dương n. Tính và in ra giá trị P = 1 + 2 + … + n
3. Tính diện tích tam giác ABC biết vị trí của 3 đỉnh trên mặt phẳng A(x1, y1), B(x2, y2) và C(x3, y3)