

# TIN ĐẠY CƯỜNG

---

## BÀI 3: DỮ LIỆU VÀ NHẬP / XUẤT DỮ LIỆU

# Nội dung buổi trước

---

- Cấu trúc của một chương trình C++:
  - Khai báo các thư viện
  - Các biến của chương trình
  - Các hàm của chương trình, trong đó hàm **main** là hàm được gọi đầu tiên khi chạy chương trình
- Khái niệm biến và quy cách đặt tên trong C++
- Khái niệm biểu thức, phép toán và phép gán
- Máy tính có 3 kiểu hoạt động chính: **tuần tự**, **lặp** và **lựa chọn**
- Khái niệm và các đặc trưng của **thuật toán**

# Nội dung chính

---

1. Một số kiểu dữ liệu thường dùng
2. Nhập và xuất dữ liệu
3. Phân rã bài toán
4. Hàm (function)
5. Ví dụ và bài tập

Phần 1

# Một số kiểu dữ liệu thường dùng

# Một số kiểu dữ liệu thường dùng

---

- Đã được đề cập đến trong bài trước: **nguyên** (int), **thực** (double, float), **logic** (bool). Dùng khi khai báo biến, hàm số,...

```
double abc; // biến abc kiểu số thực
```

```
bool xyz; // biến xyz kiểu logic
```

- **Hỏi**: tại sao cần định nghĩa “kiểu dữ liệu”?
- **Đáp**: nếu không biết kiểu của biến thì các phép toán thực hiện có thể không chính xác
  - Phép chia lấy dư chỉ thực hiện được với số nguyên
  - Kiểu logic không thể cộng với nhau

# 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 với số nguyên:
  - 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 nguyên (int)

---

```
int a = 11;
int b = 3;
cout << (a+b); // in ra 14
cout << (a-b); // in ra 8
cout << (a*b); // in ra 33
cout << (a/b); // in ra 3
cout << (a%b); // in ra 2
cout << ++a; // tăng a lên 1, in ra 12
cout << b++; // in ra 3, b tăng lên 4
cout << --b; // giảm b đi 1, in ra 3
cout << a--; // in ra 12, giảm a đi 1
```

# 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 luận lý:
  - Là kết quả của các phép so sánh:  $>$ ,  $>=$ ,  $<$ ,  $<=$ ,  $==$ ,  $!=$
  - Các phép toán logic: và ( $\&\&$ ), hoặc ( $\|\|$ ), đảo ( $!$ ), xor ( $\wedge$ )
  - Sử dụng khi ra quyết định (sẽ học ở chương 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  
cout << endl << endl; // chuyển con trỏ xuống 2 dòng
```

# 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
```

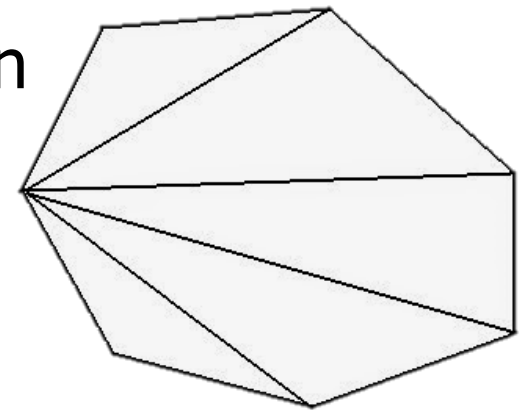
Phần 3

# Phân rã bài toán

# 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
  - Chia đa giác thành các tam giác con
  - Tính diện tích từng tam giác con
  - Lấy tổng diện tích các tam giác con



# 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)



# 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;  
}
```

# 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 lại dùng trong các bài khác
  - Bản thân các hàm toán học được sử dụng trong các bài trước là hàm do các lập trình viên khác viết ra, chúng ta chỉ sử dụng lại

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  $(x_1, y_1)$  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 }
```

# Bài tập

---

1. Nhập 2 số thực  $a$  và  $b$ . Tính  $a^b$  và in ra màn hình.  
Gợi ý: sử dụng hàm `pow` trong thư viện `cmath`
2. Tương tự như bài trên, nhưng tính  $\sqrt[b]{a}$
3. Nhập số nguyên dương  $n$ . Hãy tính và in ra màn hình giá trị  $P = 1 + 2 + \dots + n$
4. Nhập tọa độ 2 điểm trên mặt phẳng  $M(x_1, y_1)$  và  $N(x_2, y_2)$ , tính độ dài đoạn  $MN$ .
5. Tính diện tích tam giác  $ABC$  biết vị trí của 3 đỉnh trên mặt phẳng  $A(x_1, y_1)$ ,  $B(x_2, y_2)$  và  $C(x_3, y_3)$ .