



# TIN ĐẠI CƯƠNG

---

## **Bài 2: Tuần tự**



# 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 & đầu ra
- Tham trị & tham chiếu
- Có 3 loại cấu trúc điều khiển cơ bản: Tuần tự, lặp và rẽ nhánh
- Các viết, dịch, chạy và sửa lỗi chương trình máy tính



# Bài 2: Tuần tự

---

- Các khái niệm cơ sở
  - Định danh (identifier) / biến (variable)
  - Biểu thức (expression)
  - Phép gán
  - Vài kiểu dữ liệu cơ bản
- Phân rã bài toán (vấn đề)
- Hàm (function)
- Nhập và xuất dữ liệu



# Bài 2: Tuần tự

---

- Các khái niệm cơ sở
  - Định danh (identifier) / biến (variable)
  - Biểu thức (expression)
  - Phép gán
  - Vài kiểu dữ liệu cơ bản
- Phân rã bài toán (vấn đề)
- Hàm (function)
- Nhập và xuất dữ liệu

## 2.1 Các khái niệm cơ sở

```
#include <iostream>
```

thư viện  
iostream

```
using namespace std;
```

hàm  
chính

```
int main() {
```

sử dụng  
tập thư  
viện chuẩn

khai báo  
số thực x

```
double x;
```

```
cin >> x;
```

nhập x từ  
bàn phím

tính  $x^2$  và  
in ra màn  
hình

```
cout << x * x;
```

```
return 0;
```

mở đầu và  
kết thúc  
hàm chính

```
}
```

trả về 0 cho hệ thống



## 2.1.1 Định danh (identifier)

---

- Khái niệm: Vùng trong máy tính dùng để chứa những kết quả tính toán
  - Cần được đặt tên để dễ thao tác
  - Biến (variable) hay định danh
- Nguyên tắc:
  - Phải khai báo trước khi dùng
  - Phải chỉ ra kiểu (loại số)

## 2.1.1 Định danh (identifier)

- Quy tắc khai báo chung:

`<kiểu> <định danh>;`

`<kiểu> <định danh> = <giá trị>;`

- Ví dụ:

`int x;`

`int n = 100;`

`double d = 1.5;`

`double m;`

số nguyên n  
có giá trị 100

số nguyên x  
số thực m



## 2.1.2 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
- Ví dụ:

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





## 2.1.3 Phép gán

---

- Định nghĩa: Phép toán ghi kết quả vào nơi chứa dữ liệu

- Cú pháp:

**<biến> = <giá trị> ;**

- Ví dụ:

**n = 10 ;**

**m = n + 5 / 3 ;**



## 2.1.4 Vài kiểu dữ liệu cơ bản

---

- Kiểu số nguyên: int
- Kiểu số thực: double, float
- Kiểu logic: bool
  
- Các phép toán trên kiểu dữ liệu



# Kiểu nguyên (int)

---

- Dùng để lưu trữ số nguyên (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 (--)



# Kiểu thực (float, double)

---

- Dùng để lưu trữ các số thực, kiểu double có độ chính xác cao hơn float nhưng tốn nhiều bộ nhớ hơn
- Các phép tính cơ bản:
  - Các phép toán số học: cộng (+), trừ (-), nhân (\*), chia (/)
  - 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 ( $\wedge$ )
  - Sử dụng khi ra quyết định (sẽ học trong bài 4)



# Bài 2: Tuần tự

---

- Các khái niệm cơ sở
  - Định danh (identifier) / biến (variable)
  - Biểu thức (expression)
  - Phép gán
  - Vài kiểu dữ liệu cơ bản
- Phân rã bài toán (vấn đề)
- Hàm (function)
- Nhập và xuất dữ liệu



## 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 giải các bài toán con với nhau
- Ví dụ: Tính diện tích đa giác lồi
  - Chia đa giác thành các tam giác con
  - Tính diện tích các tam giác con
  - 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, bằng 0, dương) và 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)





# Bài 2: Tuần tự

---

- Các khái niệm cơ sở
  - Định danh (identifier) / biến (variable)
  - Biểu thức (expression)
  - Phép gán
  - Vài kiểu dữ liệu cơ bản
- Phân rã bài toán (vấn đề)
- Hàm (function)
- Nhập và xuất dữ liệu



## 2.3 Hàm (function)

---

- Hàm: Đoạn chương trình máy tính thực thi một thuật toán nào đó

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

```
int n = dientich(30,40);
```
- Tham chiếu & tham trị



# Bài 2: Tuần tự

---

- Các khái niệm cơ sở
  - Định danh (identifier) / biến (variable)
  - Biểu thức (expression)
  - Phép gán
  - Vài kiểu dữ liệu cơ bản
- Phân rã bài toán (vấn đề)
- Hàm (function)
- Nhập và xuất dữ liệu



# Nhập và xuất dữ liệu

---

- Xuất dữ liệu thông qua biến cout:

```
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 ghép [A = ] + giá trị a
```

- Nhập dữ liệu thông qua biến cin

```
cin >> a;              // nhập dữ liệu vào biến a  
cin >> a >> b;        // nhập kết quả vào a rồi vào b
```



# 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 }
```