

Tin học cơ sở 4

Bài 1. Giới thiệu

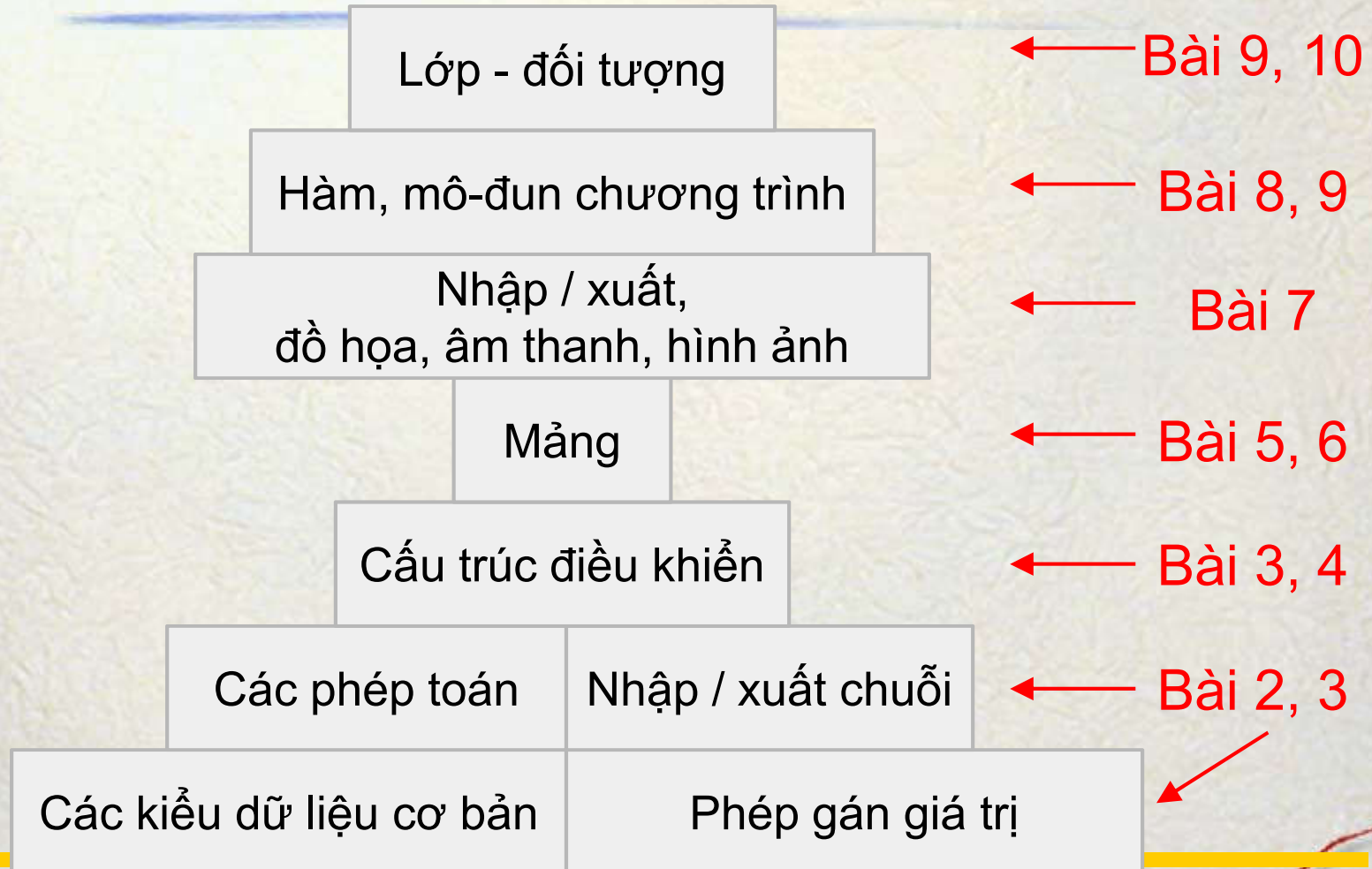
Bộ môn Khoa học máy tính - 2017



Mục tiêu khóa học

- Các khái niệm chính trong lập trình và thuật toán
- Các cấu trúc dữ liệu và thuật toán cơ bản

Các kỹ năng lập trình nền tảng



Nội dung buổi học

1. Khái niệm về chương trình và lập trình
2. Môi trường lập trình
3. Cấu trúc một chương trình
4. Biên dịch và chạy chương trình
5. Làm quen hệ thống chấm bài

Chương trình

- Một dãy các lệnh (instruction) cho máy tính
- Máy tính không thể tự vận hành
 - ❖ Cần chương trình để hoạt động
- Máy tính chạy các lệnh của chương trình tại bộ xử lý trung tâm (CPU)

Lập trình

- Quá trình tạo ra chương trình giải quyết vấn đề cụ thể bằng máy tính

Phân
tích
vấn đề

Phát
triển
thuật
toán

Đánh
giá
thuật
toán

Cài
đặt
thuật
toán

Kiểm
thử,
gỡ lỗi

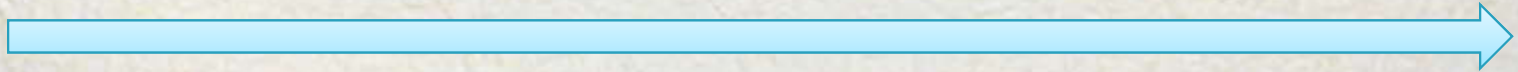
Tại sao cần lập trình

Điều
 khiển
 máy tính

Giải
 quyết
 nhiều
 vấn đề
 hiện đại

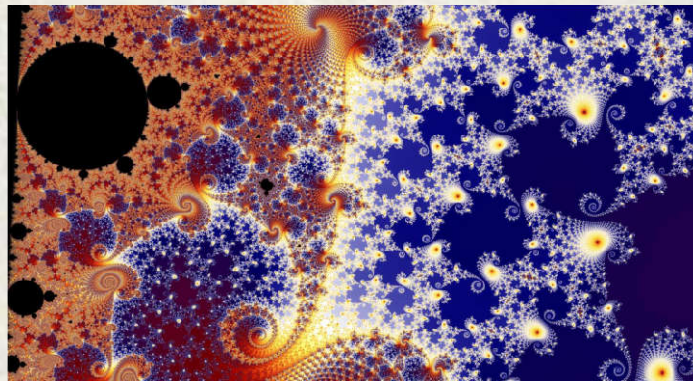
Trải
 nghiệm
 sáng tạo
 thú vị

Khai phá
 tiềm
 năng trí
 tuệ



Tại sao cần lập trình

- Sử dụng lại các phần mềm có sẵn: tốt nhưng bị giới hạn
- Tự lập trình: có thể *làm gì tùy thích*

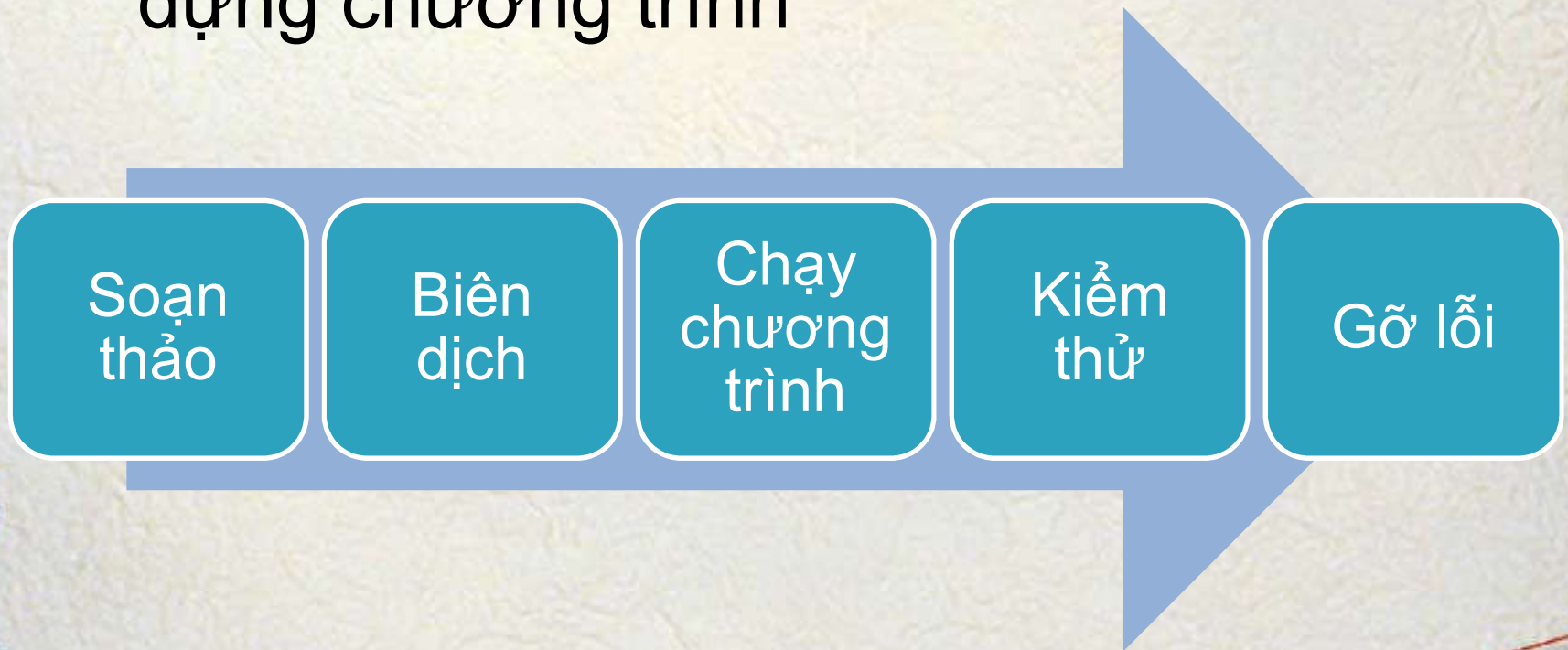


Ngôn ngữ lập trình

- ❖ Ngôn ngữ máy: khó hiểu, dễ sinh lỗi
- ❖ Ngôn ngữ tự nhiên: nhập nhằng với máy
- ❖ Ngôn ngữ bậc cao
 - Có thể chuyển thành ngôn ngữ máy
 - Sử dụng công cụ, chương trình dịch
 - Gần với ngôn ngữ tự nhiên
 - Dễ đọc, dễ trao đổi
 - C/C++, Java, Python, ...

Môi trường lập trình

- Các công cụ giúp lập trình viên xây dựng chương trình



Môi trường lập trình

- Môi trường phát triển tích hợp (IDE)
 - ❖ Phần mềm tích hợp tất cả các tính năng
- Trong khóa học này: **CodeBlocks**
- Học viên cũng có thể
 - ❖ Phối hợp các công cụ riêng lẻ
 - ❖ Thực hiện từng bước hoặc cả gói

Đánh giá chương trình



Tính đúng (correctness)



Tính chịu lỗi (robustness)



Tính dễ dùng (usability)



Tính khả chuyển (portability)



Tính khả trì (maintainability)



Tính hiệu quả (efficiency)

Chương trình đầu tiên

➤ Yêu cầu: máy tính in ra Hello, World !!!

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello, World !!!" << endl;
    return 0;
}
```

Chương trình đầu tiên

- Yêu cầu: máy tính in ra câu chào Hello, World !!!

```
#include <iostream>
```

Khai báo thư viện

```
using namespace std;
```

Khai báo không gian tên

```
int main()
```

Khai báo hàm main()

```
{
```

```
    cout << "Hello, World !!!" << endl;
```

```
    return 0;
```

câu lệnh in và xuống dòng

```
}
```

Mở đầu,
kết thúc
khối lệnh

Khai báo thư viện

- Thư viện = tập hợp các cấu trúc lập trình đã được viết sẵn
 - ❖ Dữ liệu, tài liệu, mô-đun, hàm, mã lệnh
 - ❖ Ai viết ?

```
#include <iostream>
```

Sử dụng thư viện **<iostream>**

- Nhập / xuất dữ liệu
 - ☐ In ra màn hình
 - ☐ Đọc từ bàn phím
 - ☐ ...

Khai báo không gian tên

- Namespace: tập hợp các tên

 - ❖ Tên hàm, tên biến, tên lớp ...

- Giúp tổ chức chương trình

 - ❖ Phân biệt các tên trùng nhau

Sử dụng không gian tên **std**

```
using namespace std;
```

- Chứa các tên **cout**, **endl**

- Nếu không có cần viết

 - ❑ **std::cout**

 - ❑ **std::endl**

Hàm main()

- Điểm vào của chương trình
 - ❖ Mã lệnh trong hàm **main()** được chạy trước tiên
- Chỉ có một hàm **main()** trong chương trình

```
int main()
```

Khai báo hàm main()

- **int**: kết quả là số nguyên
- **main**: tên hàm
- **()**: đây là tên một hàm
- Hàm không có tham số

Câu lệnh in ra màn hình

```
cout << "Hello, World !!!" << endl;
```

↑
Đối tượng
cout đại diện
cho luồng
xuất chuẩn

↖
Toán tử đẩy
trái: ghi ra
luồng xuất
chuẩn

↖
Dữ liệu cần ghi
là một chuỗi ký
tự: trong 2 dấu
nháy kép

↑
Đối tượng
endl đại diện
cho một dấu
xuống dòng

↖
Dấu chấm
phẩy báo hiệu
kết thúc câu
lệnh

Biểu thức

- **cout, "Hello, World !!!", endl:** số hạng, toán hạng
- **Dấu <<:** phép toán, toán tử
- **Giống như biểu thức**

❑ $1 + 2 + 3$

❑ $1 * 4 / 2$

Kết quả chính là
đối tượng cout

In nhiều dòng liên tục

```
cout << "Hello, World !!!" << endl;  
cout << "Introduction to Programming" << endl;
```

```
cout << "Hello, World !!!" << endl  
    << "Introduction to Programming" << endl;
```

Kết quả biểu thức
chính là đối tượng
cout

1 câu lệnh duy
nhất: Không
có dấu chấm
phẩy

Câu lệnh return

- Trả về kết quả của một hàm
 - ❖ Cho đoạn mã gọi hàm
 - ❖ Hàm **main**: hệ điều hành gọi và nhận kết quả trả về

```
return 0;
```

Trả về giá trị nguyên là **số 0**

- Do khai báo sẽ trả về số nguyên
- Mã thoát chương trình
 - ❑ **0**: chương trình không có lỗi
 - ❑ **1, -1, ...**: chương trình có lỗi

Biên dịch

➤ Trong CodeBlocks

- ❖ Soạn thảo file văn bản chương trình
- ❖ Ghi xuống đĩa thành file có đuôi **.cpp**
- ❖ Biên dịch:
 - Build / Build hoặc Ctrl + F9
- ❖ Kết quả: file chạy được trong hệ điều hành (Windows, Linux)

Câu hỏi

- Về nội dung buổi học ?
- Về nội dung khóa học ?
- Về cách tổ chức ?

Lập trình chuyên nghiệp bằng hàm

```
void printHelloWorld()  
{  
    cout << "Hello, World !!!" << endl;  
}  
  
int main()  
{  
    printHelloWorld();  
    return 0;  
}
```

Lập trình chuyên nghiệp bằng hàm

```
void printHelloWorld()
```

 Khai báo hàm

```
{  
    cout << "Hello, World !!!" << endl;  
}
```

Chuyển lệnh in vào trong hàm

```
int main()
```

```
{  
    printHelloWorld();  
    return 0;  
}
```

Gọi hàm

Lập trình chuyên nghiệp bằng hàm

```
void printHelloWorld()
```

Khai báo hàm printHelloWorld()

- **void**: hàm không trả về kết quả
- **printHelloWorld**: tên hàm
(đặt tên cho đoạn mã của hàm)
- Hàm không có tham số

Lập trình chuyên nghiệp bằng hàm

```
printHelloWorld();
```

Gọi hàm `printHelloWorld()`

- Chuyển quyền điều khiển cho đoạn mã trong hàm `printHelloWorld()`
- In câu Hello World
- Kết thúc hàm
- Chuyển quyền điều khiển cho đoạn mã gọi hàm trong hàm `main()`
- Thực hiện lệnh tiếp theo