

# TÓM LƯỢC BÀI GIẢNG NHẬP MÔN LẬP TRÌNH

(Vũ Quốc Hoàng, [vqhoang@fit.hcmus.edu.vn](mailto:vqhoang@fit.hcmus.edu.vn), FIT-HCMUS, 2020)

## BÀI 1 LÀM QUEN LẬP TRÌNH

### Chủ đề

- Giới thiệu lập trình, chương trình và các khái niệm cơ bản liên quan
- Khuôn chương trình đơn giản
- Biến và lệnh gán
- Nhập/xuất đơn giản
- Lỗi và phong cách viết

### Tài liệu

- [1] Vũ Quốc Hoàng, *Bí kíp luyện Lập trình C (Quyển 1)*, hBook, 2017.
- [2] Tony Gaddis, *Starting out with C++ From Control Structures through Objects*, Pearson, 8<sup>th</sup> edition, 2015.
- [3] Vũ Quốc Hoàng, *Bí kíp luyện Lập trình nhập môn với Python*, hBook, 2020.
- Đọc kĩ: Bài 1.1 [1]; Bài 1 [3]
  - Đọc thêm: Bài 1.2 [1]; Bài 2 [3]; Phần 1.1, 1.2, 1.3, 2.1, 2.2 [2]

### Kiến thức

- **Chương trình** (program) là dãy các **lệnh** (statement) viết theo một **ngôn ngữ lập trình** (programming language) (như Python, C, C++) để mô tả một **công việc** (task) cho **máy tính** (computer) **thực thi** (execute, run). Điều này tương đương với văn bản là dãy các câu viết theo một ngôn ngữ tự nhiên (như Tiếng Việt, Tiếng Anh) để mô tả một nội dung cho con người đọc hiểu. **Lập trình** (programming) là việc viết chương trình và **lập trình viên** (programmer) là người viết chương trình.
- Thuật ngữ **mã nguồn** (source code, code) thường được dùng để chỉ mặt hình thức (văn bản) của chương trình hơn là nội dung (công việc). Mã nguồn thường được lưu trữ trong các **tập tin mã nguồn** (source code file) có phần mở rộng là .py, .c, .cpp cho Python, C, C++.
- Chương trình, khi được máy thực thi, sẽ tương tác với **người dùng** (user), mà thường thì chương trình nhận thông tin **đầu vào** (input) từ người dùng và xuất ra các **kết xuất** (output) phù hợp cho người dùng.
- Có nhiều loại lệnh khác nhau, chẳng hạn, **lệnh xuất** (output statement) yêu cầu máy xuất ra một chuỗi nào đó cho người dùng, **lệnh nhập** (input statement) yêu cầu máy nhập một chuỗi nào đó từ người dùng (người dùng sẽ gõ chuỗi từ bàn phím và nhấn Enter để nhập).
- **Chuỗi** (string) là dãy các kí tự, được mô tả trong chương trình bằng cặp nháy kép ("...") gọi là **hằng chuỗi** (string literal, string constant). Chuỗi cũng có thể được nhập vào từ người dùng.

- **Khuôn chương trình** là bộ khung hay mẫu chương trình gồm các phần cố định và các phần thay đổi. Lúc mới học lập trình, ta nên thuộc vài khuôn chương trình đơn giản và chỉnh sửa phần thay đổi theo nhu cầu. Mẫu 1 là khuôn chương trình đơn giản nhất, chỉ xuất. Mẫu 2 là khuôn chương trình có tương tác với người dùng, xuất-nhập-xuất (hay hỏi-đáp).
- **Biến** (variable) là phương tiện để nhớ, tức là để lưu trữ **thông tin** (information) hay **dữ liệu** (data), của chương trình. Ta cần biến để chứa dữ liệu nhập từ người dùng (và các mục đích khác). Dữ liệu cũng có thể được “đưa” vào biến bằng **lệnh gán** (assignment statement).
- Một số ngôn ngữ (như C, C++) yêu cầu phải **khai báo biến** (variable declaration), tức là báo cho biết **tên** (name), còn gọi là **định danh** (identifier), và **kiểu dữ liệu** (data type) mà biến sẽ chứa, trước khi dùng biến. Một số ngôn ngữ (như Python) thì không cần.
- Tên biến phải được đặt đúng qui tắc và nên ngắn gọn, rõ ràng, gợi nhớ đến ý nghĩa của biến. Đa số các ngôn ngữ (như Python, C, C++) **phân biệt chữ hoa chữ thường** (case sensitive).
- Biến dùng để chứa dữ liệu nhập được gọi là **biến nhập** (input variable) và nó thường là biến chuỗi (biến chứa chuỗi) vì người dùng thường nhập chuỗi. Biến chứa dữ liệu được xuất ra (cũng thường là biến chuỗi) được gọi là **biến xuất** (output variable).
- **Ghi chú** (comment) là những chú thích, giải thích, mô tả thêm trong mã nguồn. Chúng không tham gia vào chương trình mà giúp việc đọc hiểu mã nguồn dễ hơn. Ghi chú có thể ngắn trên một dòng (từ dấu # trong Python, dấu // trong C/C++ đến hết dòng) hoặc dài trên nhiều dòng (giữa cặp dấu /\* ... \*/ trong C/C++).
- Ta cần cài đặt **môi trường phát triển tích hợp** (Integrated development environment, IDE), là bộ các công cụ, để lập trình với ngôn ngữ nào đó. Có nhiều IDE, lúc mới học nên dùng IDLE cho Python và Dev C++ cho C/C++.
- Các bước đơn giản để **phát triển** (develop) một chương trình là: (1) tạo file; (2) viết mã nguồn và lưu file; (3) kiểm tra lỗi; (4) chạy; (5) dùng thử. Với các **ngôn ngữ biên dịch** (compiled language) (như C, C++) thì có thêm bước **biên dịch** (compile) giữa (3) và (4) để tạo **chương trình chạy được** (executable program) (như các file .exe trên Windows), với các **ngôn ngữ thông dịch** (interpreted language) (như Python) thì không cần. Bước (3) và (4) (cùng với bước biên dịch nếu có) thường được IDE gom vào 1 thao tác (như F5 trong IDLE và F11 trong Dev C++).
- Chương trình có **lỗi cú pháp** (syntax error) khi viết không đúng theo qui định của ngôn ngữ, khi đó, nó không chạy được (hay không biên dịch được). Các IDE hỗ trợ việc kiểm tra lỗi cú pháp và thông báo trợ giúp sửa lỗi.
- Chương trình chạy được (không có lỗi cú pháp) vẫn có thể có lỗi lúc chạy, gọi là **lỗi thực thi** hay **lỗi lúc chạy** (runtime error).
- Ta không chỉ cần viết chương trình cho đúng mà nên viết cho đẹp, ngăn nắp, rõ ràng và tuân theo **phong cách viết** (style) chung mà đa số các lập trình viên hay dùng. Cần nghiêm túc rèn luyện việc này (chẳng hạn, bằng cách bắt chước các mã nguồn minh họa trong môn học) khi mới học lập trình để tránh các thói quen xấu khó sửa sau này.

## Kĩ năng

- Cài đặt được các IDE “nhỏ-nhẹ”
- Thành thạo các bước phát triển một chương trình
- Biết cách sửa lỗi khi chương trình bị lỗi cú pháp
- Viết được các chương trình đơn giản (dạng “chỉ xuất” và “hỏi-đáp”)
- Ý thức và rèn luyện được việc viết chương trình đẹp, ngăn nắp, rõ ràng

## Mã nguồn minh họa

Mã nguồn 1. (Khuôn chương trình chỉ xuất)

Tiếng Việt	Tiếng Python
Chào mọi người! Tôi là Hoàng.	<pre>print("Hello everyone!") print("I am a Python program")</pre>
Tiếng C	Tiếng C++
<pre>#include &lt;stdio.h&gt;  int main() {     printf("Hello everyone!\n");     printf("I am a C program");      return 0; }</pre>	<pre>#include &lt;iostream&gt; using namespace std;  int main() {     cout &lt;&lt; "Hello everyone!\n";     cout &lt;&lt; "I am a C++ program";      return 0; }</pre>
<b>Giải thích và bình luận thêm:</b> <ul style="list-style-type: none"><li>- Văn bản trong ngôn ngữ tự nhiên (Tiếng Việt, Tiếng Anh) dùng để giao tiếp; chương trình trong ngôn ngữ lập trình (Python, C, C++) dùng để yêu cầu máy tính thực thi các công việc, cho nên câu trong chương trình được gọi là câu lệnh (hay lệnh)</li><li>- Mức độ gần gũi ngôn ngữ tự nhiên (còn gọi là mức cao, high-level) là: Python &gt; C++ &gt; C</li><li>- C++ rất giống C; C++ là một mở rộng, phát triển, cải tiến của C; (hầu hết các chương trình C có thể được xem là chương trình C++)</li><li>- C/C++: kết thúc lệnh bắt buộc có dấu chấm phẩy (;); Python: không bắt buộc nhưng nếu viết trên một dòng thì phải có dấu chấm phẩy phân cách các lệnh</li><li>- Muốn xuất nhiều chuỗi ta dùng nhiều lệnh xuất hoặc dùng chuỗi "lớn" gồm nhiều dòng, kí hiệu đặc biệt \n là xuống dòng (trong C++ có thể dùng endl)</li><li>- Chuỗi trong Python cũng có thể để trong cặp nháy đơn ( '...' ) nhưng nên để trong cặp nháy kép ("...") như C/C++ (và nhiều ngôn ngữ lập trình khác)</li><li>- Thử bỏ dấu chấm phẩy (C/C++) hay dấu đóng mở ngoặc tròn thì chương trình bị lỗi cú pháp, không chạy được, do đó phải viết đúng cú pháp để chương trình chạy được</li><li>- Thử bỏ/thêm các khoảng trắng (bên ngoài chuỗi), xuống dòng, thụt đầu dòng, ... chương trình vẫn chạy được nhưng mã nguồn khó đọc, do đó nên trình bày đẹp để chương trình dễ đọc; dùng phím Tab để thụt đầu dòng</li><li>- Mới học lập trình nên thuộc các khuôn chương trình và chỉ tập trung vào phần thay đổi (phần đóng khung đỏ trong hình) để không bị rối hay ngợp</li><li>- printf là viết tắt của "print formatted"; cout là viết tắt của "console output"; chữ n trong \n là viết tắt của "new line"</li><li>- Nếu muốn chương trình xuất tiếng (giống như nói) thay vì xuất chuỗi (giống như viết) thì sao?</li></ul>	

Mã nguồn 2. (Khuôn chương trình "hỏi-đáp")

Tiếng Anh	Python
A: What's your name? B: Hoang. A: Hello Hoang.	<pre># A: chương trình, B : người dùng print("Bạn tên gì?") tên = input() print("Chào", tên)</pre>

C	C++
<pre>// A: program, B: user #include &lt;stdio.h&gt;  int main() {     char name[100];      printf("What's your name? ");     gets(name);     printf("Hello %s", name);      return 0; }</pre>	<pre>// A: program, B: user #include &lt;iostream&gt; using namespace std;  int main() {     string name;      cout &lt;&lt; "What's your name? ";     cin &gt;&gt; name;     cout &lt;&lt; "Hello " &lt;&lt; name;      return 0; }</pre>
<p><b>Giải thích và bình luận thêm:</b></p> <ul style="list-style-type: none"> <li>- Ghi chú trong Python bắt đầu bằng dấu #, C/C++ bắt đầu bằng dấu //; trong C/C++ có thể dùng ghi chú trên nhiều dòng với cặp dấu /* ... */</li> <li>- Hoạt động giao tiếp thường gồm nhiều người tương tác với nhau; hoạt động của chương trình gồm chương trình và người dùng tương tác với chương trình; dạng tương tác đơn giản nhất là hỏi-đáp</li> <li>- Để có thể nhận thông tin từ người dùng (hay tổng quát là để nhớ), chương trình cần dùng biến</li> <li>- Biến trong Python không cần khai báo, biến trong C/C++ cần khai báo để biết tên và kiểu</li> <li>- C chi tiết/cụ thể hơn C++ nên cần nói rõ biến name sẽ chứa dãy (mảng) các kí tự và cụ thể bao nhiêu chứ không chỉ nói chung chung là chuỗi (string) như C++</li> <li>- Lệnh gán/lệnh nhập giúp đặt giá trị vào biến, sau đó nó có thể được dùng (chẳng hạn để xuất trong lệnh xuất)</li> <li>- Ta gọi chuỗi mô tả trong "..." là hằng chuỗi (vì nó cố định) để phân biệt với chuỗi chứa trong biến chuỗi</li> <li>- Python hỗ trợ các kí tự Tiếng Việt (và các tiếng khác) rất tốt so với C/C++</li> <li>- Trong mã nguồn C, thử sửa con số trong khai báo chuỗi nhỏ lại (5 chẳng hạn) và lúc chạy gõ tên dài dài (20 kí tự chẳng hạn) thì chương trình bị lỗi thực thi (chương trình “quay quay” và báo “return value” khác 0)</li> <li>- char là viết tắt của character; gets là viết tắt của “get string”; cin là viết tắt của “console input”; chữ s trong %s (ở lệnh xuất của C) là viết tắt của string</li> <li>- Giả sử người dùng không trả lời ngắn gọn chỉ cái tên (như “Hoàng”) mà là câu đầy đủ hơn (như “Tên tôi là Hoàng” hay “My name is Hoang”) thì sao?</li> </ul>	

### Bài tập

1. Cài đặt và làm quen các công cụ lập trình (IDE): IDLE cho Python, Dev C++ cho C/C++.
2. Làm các bài tập sau bằng C, C++: Bài 1.1 [1], 1.2.1, 1.2.2 [1].
3. Làm thêm các bài tập sau bằng Python: Bài 1 [3].