

Chương 1

Giới thiệu về lập trình

*Khoa công nghệ thông tin
Trường Đại học mở tphcm*

1

Mục tiêu

- Cung cấp các khái niệm cơ bản: lập trình, chương trình, ngôn ngữ lập trình, chương trình dịch, môi trường lập trình.
- Các bước xây dựng một chương trình đơn giản.
- Cách dùng ngôn ngữ tự nhiên và lưu đồ để mô tả thuật giải của một chương trình.
- Các thành phần cơ bản của một chương trình C++.
- Cách tạo và thực thi một chương trình C++ trong môi trường lập trình Visual Studio.

2

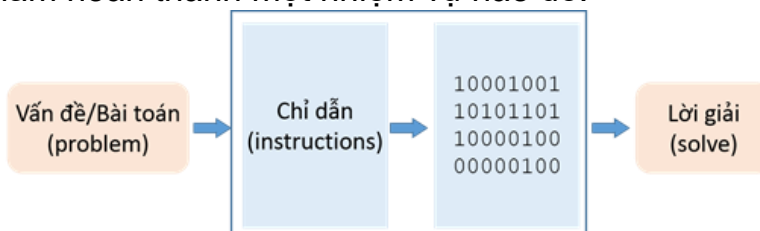
Nội dung

1. Một số khái niệm
2. Khái quát về ngôn ngữ C++
3. Các thành phần cơ bản của một chương trình C++
4. Tạo và thực thi chương trình C++
5. Các bước xây dựng chương trình

3

1.1 Một số khái niệm

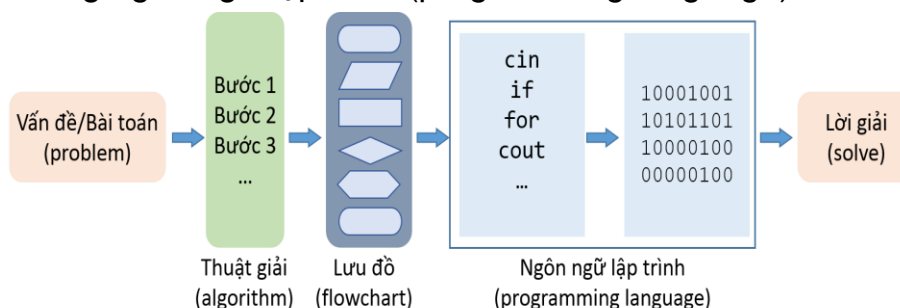
- **Máy tính** (computer) là máy lưu trữ dữ liệu (số, văn bản, hình ảnh), tương tác với các thiết bị (màn hình, máy in, loa) và thực thi chương trình.
- **Chương trình máy tính** (computer program) là một chuỗi các chỉ dẫn/lệnh (instructions) để máy tính thực hiện nhằm hoàn thành một nhiệm vụ nào đó.



4

Một số khái niệm

- **Lập trình** (computer programming) là thiết kế (designing) và hiện thực (implementing) các chương trình máy tính bằng ngôn ngữ lập trình (programming language).



5

Một số khái niệm

- **Ngôn ngữ lập trình**
 - Ngôn ngữ máy (machine language/machine code)
 - Ngôn ngữ assembly (assembly language)
 - Ngôn ngữ cấp cao (high-level language)
 - COBOL (Common Common Business-Oriented Language), BASIC (Beginner's All-Purpose Symbolic), C...
 - C++, C#, Java, Python, Ruby, Perl...

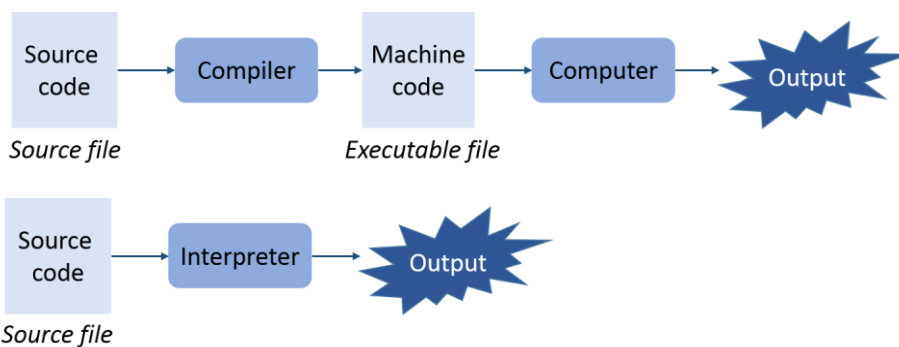
6

Một số khái niệm

- Các thành phần cơ bản của một ngôn ngữ lập trình
 - **Biến** (variables): là vùng nhớ dùng để lưu trữ một giá trị dữ liệu, mỗi biến có một tên.
 - **Cấu trúc điều khiển** (control structures/logic structures): điều khiển trật tự thực hiện các câu lệnh của chương trình (flow control).
 - **Cấu trúc dữ liệu** (data structures): cách tổ chức và lưu trữ dữ liệu trong máy tính để sử dụng dữ liệu có hiệu quả.
 - **Cú pháp** (syntax): tập hợp các quy tắc của một ngôn ngữ lập trình mà người lập trình cần phải tuân thủ.
 - **Công cụ** (tools): là môi trường giúp người lập trình phát triển chương trình (Integrated Development Environment - IDE).

Một số khái niệm

- Chương trình dịch
 - **Chương trình nguồn** (source program) hay mã nguồn (source code) được viết bằng một ngôn ngữ lập trình cấp cao và được lưu trong một file (văn bản không định dạng - plain text).
 - **Chương trình dịch** sẽ chuyển chương trình nguồn sang mã máy (machine code) để máy tính thực thi.
 - **Biên dịch** (compiler): chương trình dịch đọc toàn bộ mã nguồn của một chương trình và chuyển tất cả sang mã máy, sau đó máy tính sẽ thực hiện mã máy.
 - **Thông dịch** (interpreter): chương trình dịch đọc mã nguồn và thực hiện từng dòng lệnh của mã nguồn đó.



9

1.2 Khái quát về ngôn ngữ C++

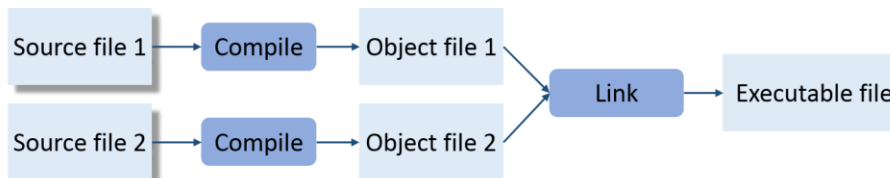
- C++ do Bjarne Stroustrup phát minh năm 1979 tại AT&T Bell Laboratories, dựa trên C.
- C++ là một ngôn ngữ lập trình:
 - hỗ trợ trừu tượng hóa dữ liệu (data abstraction)
 - hỗ trợ lập trình hướng đối tượng (object-oriented programming),
 - hỗ trợ lập trình tổng quát (generic programming).



10

Khái quát về ngôn ngữ C++

- C++ là ngôn ngữ biên dịch. Chương trình nguồn được trình biên dịch chuyển sang mã đối tượng (mã máy), sau đó liên kết với các mã đối tượng khác (thư viện) để tạo thành chương trình thực thi.



- C++ là ngôn ngữ phân biệt ký tự hoa và thường (case sensitive).
 - someName khác với SomeName

11

1.3 Các thành phần cơ bản của chương trình C++

- // Chương trình đầu tiên: hello.cpp

Chú thích trên nhiều dòng, đặt giữa /* và */

```

/* Hello.cpp : This file contains the 'main' function.
Program execution begins and ends there.*/
#include <iostream>
using namespace std;
int main()
{
    //in ra màn hình dòng Hello World!
    cout << "Hello World!\n";
    return 0;
}
  
```

Chỉ thị tiền xử lý (preprocessor directive)

hàm main

Khai báo namespace, sử dụng cout thay vì std::cout

Chú thích trên một dòng, đặt sau //

Chương trình hello.cpp chỉ có một hàm int main(): tên hàm là main
(): sinh sách tham số trống
kiểu trả về là int
Thân hàm có hai câu lệnh.

12

Các thành phần cơ bản của chương trình C++

- Chú thích/ghi chú (comment) dùng để mô tả, giải thích, trình biên dịch sẽ bỏ qua phần này
 - trên một dòng: đặt sau //
 - trên nhiều dòng: đặt giữa /* */
- Chỉ thị tiền xử lý (preprocessor directive):
 - Bắt đầu bằng ký tự #.
 - `#include <iostream>`: sử dụng các đối tượng nhập/xuất như cin, cout,... có trong iostream (header file).
- Namespace
 - `using namespace std;`
 - Sử dụng tên các đối tượng có trong vùng tên chuẩn std như các tên cin, cout
 - Nếu không khai báo using namespace std; thì khi dùng cout phải ghi rõ: std::cout.

13

Các thành phần cơ bản của chương trình C++

- Hàm

```
#include <iostream>
using namespace std;
int main()
{
    //in ra màn hình dòng Hello World!
    cout << "Hello World!\n";
    return 0;
}
```

- Chương trình C++ có thể có nhiều hàm (function) và chương trình bắt đầu thực hiện từ hàm main.
- Mỗi hàm có:
 - Kiểu trả về.
 - Tên hàm.
 - Danh sách tham số đặt giữa hai dấu ngoặc đơn (). Thân hàm đặt giữa hai dấu ngoặc nhọn {}, liệt kê các câu lệnh/phát biểu mà hàm thực hiện.

14

Các thành phần cơ bản của chương trình C++

• Câu lệnh:

```
cout << "Hello, world!\n";
```

- in ra màn hình dòng chữ **Hello, world!**
- '\n' là ký tự xuống hàng
- **cout** được khai báo trong iostream (namespace std).
- Chuỗi xuất ra màn hình được đặt giữa hai dấu nháy kép.
- Có thể dùng ký hiệu **endl** để xuống hàng thay cho '\n':

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

```
return 0;
```

- Câu lệnh trả về (return statement) giá trị 0, báo cho hệ thống chương trình kết thúc thành công.
- Câu lệnh (statement) kết thúc bằng dấu chấm phẩy (;).

15

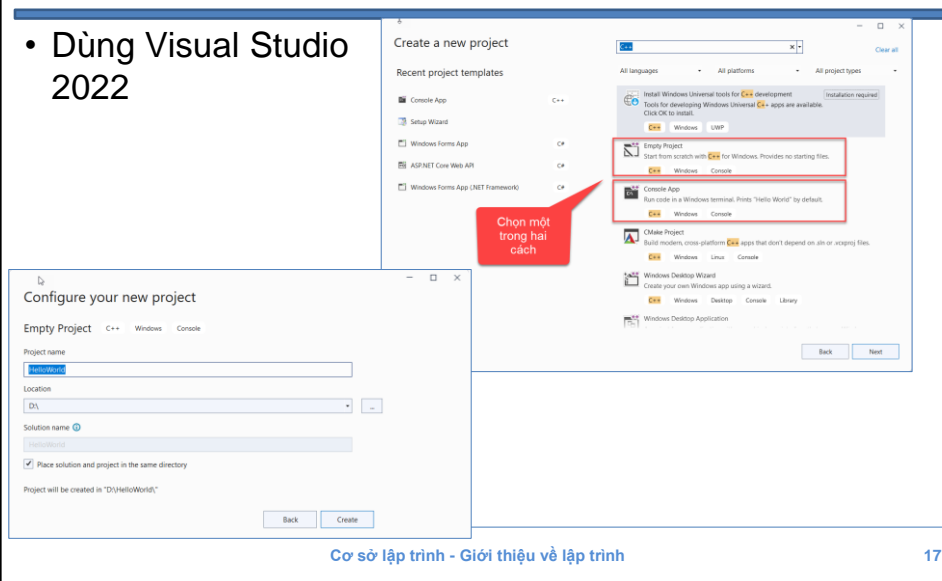
Bài tập

- Viết chương trình in ra màn hình họ và tên của bạn.
- Viết chương trình in ra màn hình "Ban có thích lập trình không?".
- Viết chương trình in ra màn hình mã số sinh viên và họ tên của bạn trên hai dòng.

16

1.4 Tạo và thực thi chương trình C++

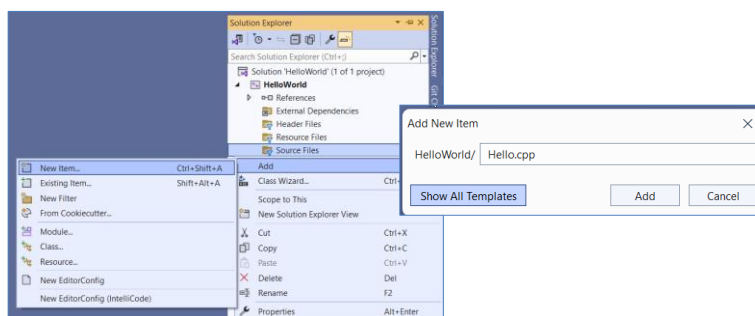
- Dùng Visual Studio 2022



17

1.4 Tạo và thực thi chương trình C++

- Thêm file cpp vào thư mục Source Files

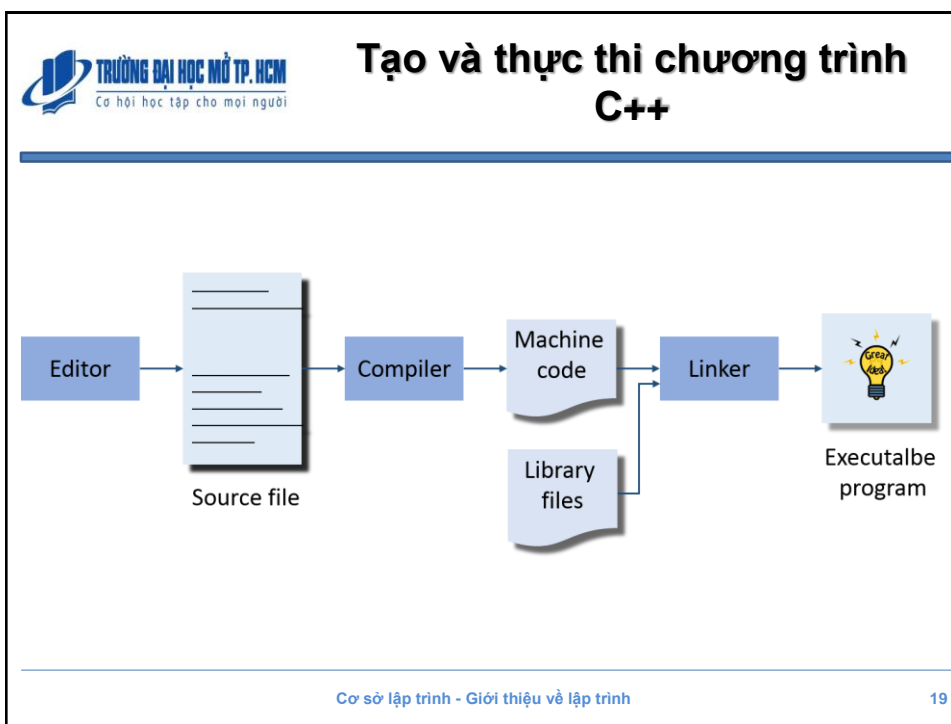


- Viết code
- Biên dịch: Ctrl + F7 (Build → Compile)
- Thực thi: F5/Ctrl + F5
 - (Debug → Start Debugging/Start Without Debugging)

Cơ sở lập trình - Giới thiệu về lập trình

18

18



19

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Xử lý lỗi

- Lỗi do compiler phát hiện: **compile-time errors**.
- Lỗi do linker phát hiện: **link-time errors**.
- Lỗi khi chương trình thực thi: **run-time errors** hoặc **logic errors**.
- Ví dụ 1:
 - Không `#include` hoặc thiếu `using namespace std`

```

//no #include here
int main()
{
    cout << "Hello, world!\n";
    return 0;
}
      
```

→ error: 'cout' : undeclared identifier

Cơ sở lập trình - Giới thiệu về lập trình

20

20

- Ví dụ 2: gõ sai iostram (thay vì iostream)

```
#include <iostram>
using namespace std;
int main()
{
    cout << "Hello, world!\n";
    return 0;
}
```

→ error: Cannot open include file: 'iostram': No such file or directory

→ error: 'cout' : undeclared identifier

21

- Ví dụ 3: thiếu dấu nhảy đôi kết thúc chuỗi

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello, world!\n";
    return 0;
}
```

→ error: expected a ';'

22

- Ví dụ 4: Gõ sai từ khóa int

```
#include <iostream>
using namespace std;
integer main()
{
    cout << "Hello, world!\n";
    return 0;
}
```

→ error: identifier "integer" is undefined

- Ví dụ 5: Dùng < thay vì <<

```
#include <iostream>
using namespace std;
int main()
{
    cout < "Hello, world!\n";
    return 0;
}
```

→ warning: '<' : operator has no effect; expected operator with side-effect

- Ví dụ 6: dùng dấu nháy đơn thay vì dấu nháy đôi

```
#include <iostream>
using namespace std;
int main()
{
    cout << 'Hello, world!\n';
    return 0;
}
```

→ error: too many characters in constant

- Ví dụ 6: dùng dấu nháy đơn thay vì dấu nháy đôi

```
#include <iostream>
using namespace std;
int main()
{
    cout << 'Hello, world!\n';
    return 0;
}
```

→ error: too many characters in constant

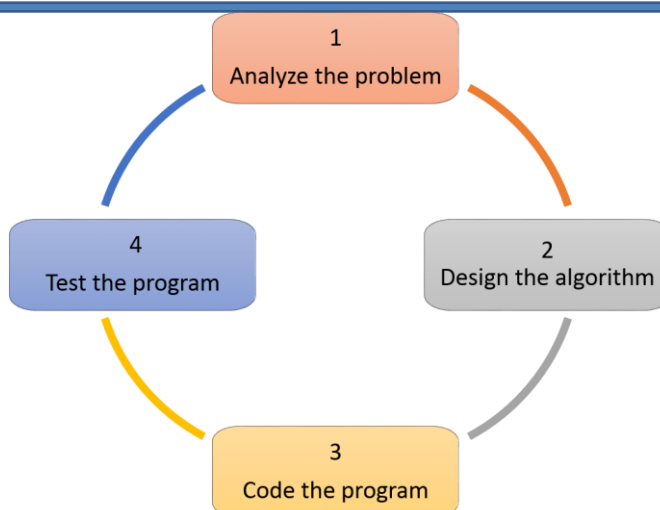
- Ví dụ 7: Thiếu dấu chấm phẩy ;

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello, world!\n"
    return 0;
}
```

→ error: missing ';' before 'return'

27

1.5 Các bước xây dựng chương trình



28

Các bước xây dựng chương trình

- **Phân tích vấn đề (analyze the problem):** Xác định dữ liệu sẽ được cung cấp (input) và kết quả cần phải có (output).
 - Output: mục tiêu của việc giải quyết vấn đề.
 - Input là các dữ liệu cần có để đạt được mục tiêu.
 - Sử dụng lưu đồ IPO (Input, Processing, Output) để sắp xếp và trình bày kết quả của bước phân tích.
- **Thiết kế thuật giải (design the algorithm):** Đây là bước trọng tâm của quy trình giải quyết vấn đề. Tùy vào tính phức tạp và khó khăn của từng vấn đề, bước này có thể chỉ cần một người làm việc trong vài giờ hoặc cần đến cả một đội ngũ lập trình viên làm việc trong vài tháng.
 - Sử dụng flowchart hoặc pseudo-code (mã giả) để mô tả thuật giải.

Các bước xây dựng chương trình

- **Lập trình (code the program):** Viết các câu lệnh (mã hóa chương trình) bằng một ngôn ngữ lập trình để hiện thực bản thiết kế đã tạo ở bước 2. Kết quả của bước này là chương trình (program).
 - Sử dụng ngôn ngữ lập trình C++.
- **Kiểm thử (test the program):** Kiểm tra có thực sự giải quyết được vấn đề đã cho hay chưa?



Bước 1. Phân tích vấn đề

- **Mô tả vấn đề:** Một cửa hàng bán ti vi cần một chương trình tính tiền và in và số tiền khách phải trả. Số tiền khách hàng phải trả được tính bằng cách lấy giá bán ti vi nhân với tỷ lệ thuế, sau đó cộng với giá bán.
 - Xác định output: tiền phải trả.
 - Xác định input: giá bán và tỷ lệ thuế.
 - Dùng lưu đồ IPO để trình bày kết quả của bước phân tích:

Input	Processing	Output
giá bán	Processing items:	tiền phải trả
tỷ lệ thuế	Algorithm:	

Cơ sở lập trình - Giới thiệu về lập trình

31

31



Bước 2. Thiết kế thuật giải

- Viết thuật giải để chuyển input thành output. Thuật giải được viết trong cột Processing của lưu đồ IPO.
- Mỗi lệnh trong thuật giải sẽ mô tả hành động mà máy tính sẽ thực hiện theo đúng thứ tự để có được output từ input. Vì thế, các lệnh thường bắt đầu bằng động từ.

Input	Processing	Output
giá bán tỷ lệ thuế	Processing items: Algorithm: 1. nhập giá bán và tỷ lệ thuế 2. tính tiền phải trả = giá bán * tỷ lệ thuế + giá bán 3. in ra tiền phải trả	tiền phải trả

Cơ sở lập trình - Giới thiệu về lập trình

32

32

Thiết kế thuật giải

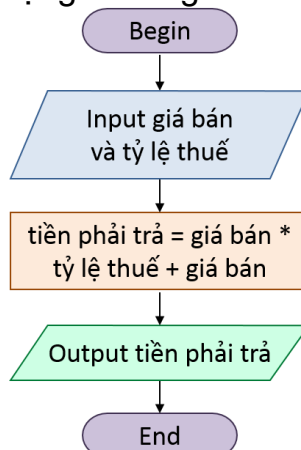
- Kiểm tra thuật giải (desk-checking/hand-tracing): dùng giấy và bút chì thực hiện từng bước của thuật giải.
- Chọn một bộ dữ liệu nhập mẫu, ví dụ:

– giá bán	tỷ lệ thuế	giá bán * tỷ lệ thuế	tiền phải trả
–2300	0.05	115	2415
–5200	0.03	156	5356
- Thực hiện kiểm tra với cả dữ liệu nhập hợp lệ (valid input) và không hợp lệ (invalid input). Dữ liệu nhập không hợp lệ là dữ liệu mà thuật giải không muốn người sử dụng nhập, ví dụ nhập giá bán là số âm.

33

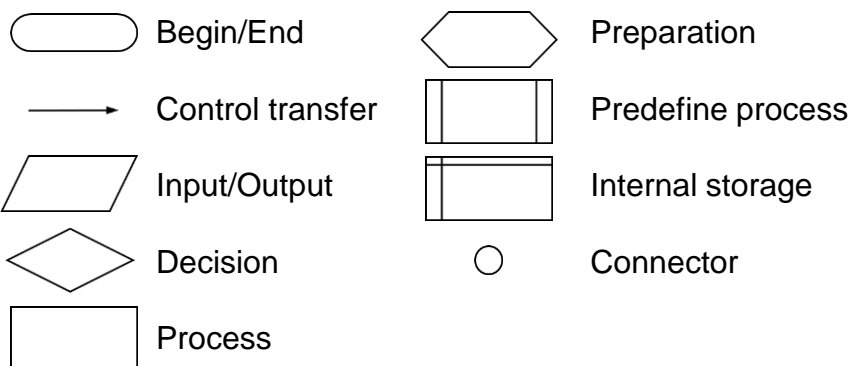
Thiết kế thuật giải

- Có thể mô tả thuật giải bằng lưu đồ:



34

- Một số ký hiệu dùng trong lưu đồ



- Sử dụng Flowgorithm (<http://www.flowgorithm.org>) để vẽ lưu đồ.

Bước 3. Lập trình

- Chuyển mã giả của thuật giải thành các câu lệnh của một ngôn ngữ lập trình.
- Trong chương trình nên có các chú thích (comment) để giải thích mục đích của các câu lệnh. Có hai loại chú thích cơ bản:
 - Header comment:** đặt ở phần đầu của chương trình (chương trình con), cung cấp thông tin tổng quát của chương trình.
 - Step comment:** còn gọi là in-line comment, đặt trong thân chương trình, giải thích mục đích của các câu lệnh.
- Chú thích được trình biên dịch bỏ qua, vì vậy không ảnh hưởng đến sự thực thi của chương trình. Chương trình được chú thích tốt sẽ giúp các lập trình viên khác hiểu chương trình một cách dễ dàng. Tuy nhiên, không nên quá lạm dụng.

```

//Chương trình tính tiền mua TV-Tac gia: NTTA
#include <iostream>
using namespace std;
int main()
{
    //khai bao bien
    double giaban = 0.0;
    double tylethue = 0.0;
    double tienphaitra = 0.0;
    //nhap du lieu
    cout << "Nhap gia ban: ";
    cin >> giaban;
    cout << "Nhap ty le thue: ";
    cin >> tylethue;
    //tinh tien khach phai tra
    tienphaitra = giaban * tylethue + giaban;
    //in ket qua
    cout << "So tien khach tra: " << tienphaitra << endl;
    return 0;
}

```

37

37



Bước 4. Kiểm thử

- Kiểm thử được thực hiện ở mỗi bước trong chu trình. Khi chương trình hoàn tất, kiểm thử được thực hiện lần cuối và cũng là lần quan trọng nhất.
- Chạy chương trình với dữ liệu nhập mẫu (test data) để so sánh kết quả chương trình với kết quả đã kiểm thử bằng tay (ở bước 2).
 - Nhập gia ban: 2300
 - Nhập ty le thue: 0.05
 - So tien khach tra: 2415
 - Nhập gia ban: 5200
 - Nhập ty le thue: 0.03
 - So tien khach tra: 5356

38

Bài tập

- Cho lưu đồ IPO như sau:

Input	Processing	Output
số lượng bán giá mua giá bán	Processing items: Algorithm: 1. nhập số lượng bán, giá mua và giá bán 2. tính tiền chênh lệch giữa giá mua và giá bán 3. tính tiền lời bằng cách nhân tiền chênh lệch với số lượng bán 4. in ra tiền lời	tiền lời

- Hãy cho biết giá trị tiền lời được in ra nếu nhập số lượng bán là 100, giá mua là 5 và giá bán là 8.
- Hãy hiệu chỉnh lưu đồ IPO để làm giảm bước tính toán.

Bài tập

- Hãy viết thuật giải cho vấn đề sau đây:
 - **Mô tả vấn đề:** Một cửa hàng bánh bán hai loại: bánh bông lan và bánh su, giá bán bằng nhau. Người quản lý cửa hàng cần có một chương trình tính tổng số bánh (bông lan và bánh su) khách hàng mua và tổng số tiền khách hàng phải trả.
 - Mô tả lưu đồ IPO.
 - Kiểm tra với dữ liệu nhập là 4, 2, 0.5 và 12, 0 và 0.45.

Q & A