Kỹ Thuật Lập Trình

(Ngôn Ngữ Lập Trình C)

Bài giảng số 1

Giới thiệu về môn học

GV. Nguyễn Trí Cường

C9-101, email: cuong.nguyentri@hust.edu.vn

Sđt: 0983309963

- Teams, bảo mật
- Giờ: trc 7h: 9:30
- Giữ im lặng
  - Vi phạm: mời ra ngoài + ghi tên trừ 1 điểm/lần
- Ko điểm danh
- 2-1-0-4:
  - 2: lý thuyết + chữa BT.
  - 1: chữa BT + BTVN (có tính điểm)
- Thi: giữa kỳ (có điểm công và trừ) + cuối kỳ (chỉ tính bài thi)
  - Chỉ thi trong nội dung học và BT

Mã máy

- Máy tính chỉ nhận các tín hiệu điện tử - có, không có -tương ứng với các dòng bits.
- 1 program ở dạng đó gọi là machine code.
- Ban đầu chúng ta phải dùng machine code để viết CT:
- Quá phức tạp, giải quyết các bài toán lớn là không tưởng

```
23fc 0000 0001 0000 0040
0cb9 0000 000a 0000 0040
6e0c
06b9 0000 0001 0000 0040
60e8
```

ASSEMBLY LANGUAGE

- NN Assembly là bước đầu tiên của việc xây dựng cơ chế viết chương trình tiện lợi hơn – thông qua các ký hiệu, từ khóa và cả mã máy.
- Tất nhiên, để chạy được các chương trình này thì phải dịch (assembled) thành machine code.
- Vẫn còn phức tạp, cải thiện không đáng kể

```
   movl   #0x1,n
compare:
   cmpl   #oxa,n
   cgt    end_of_loop
   acddl  #0x1,n
   bra    compare
end_of_loop:
```

Ngôn ngữ lập trình cấp cao

- Thay vì dựa trên phần cứng (machine-oriented) cần tìm cơ chế dựa trên vấn đề (problem-oriented) để tạo chương trình.
- Chính vì thế *high(er) level* languages – là các ngôn ngữ lập trình gần với ngôn ngữ con người hơn – dùng các từ khóa tiếng anh – đã được xây dựng như : Algol, Fortran, Pascal, Basic, Ada, C, …

Chương trình máy tính

Những tính chất cần có với các chương trình phần mềm

- Tính mềm dẻo scalability / Khả năng chỉnh sửa modifiability
- Khả năng tích hợp integrability / Khả năng tái sử dụng reusability
- Tính chuyển đổi, linh hoạt, độc lập phần cứng -portability
- Hiệu năng cao -performance
- Độ tin cậy - reliability
- Dễ xây dựng
- Rõ ràng, dễ hiểu
- Ngắn gọn, xúc tích

Lịch sử ra đời

- 1940s : Machine code
- 1950s Khai thác sức mạnh của MT: Assembler code, Autocodes, first version of Fortran
- 1960s Tăng khả năng tính toán: Cobol, Lisp, Algol 60, Basic, PL/1 --- nhưng vẫn dùng phong cách lập trình cơ bản của assembly language.
- 1970s Bắt đầu cuộc khủng hoảng phần mềm “software crisis”:

  1. Giảm sự phụ thuộc vào máy – Tính chuyển đổi.
  2. Tăng sự đúng đắn của chương trình -Structured Programming, modular programming và information hiding.

  Ví dụ : Pascal, Algol 68 and C.

Lịch sử ra đời

- 1980s Giảm sự phức tạp – object orientation, functional programming.
- 1990s Khai thác phần cứng song song và phân tán (parallel và distributed) làm cho chương trình chạy nhanh hơn, kết quả là hàng loạt ngôn ngữ mở rộng khả năng lập trình parallel
- 2000s Ngôn ngữ lập trình phục vụ phát triển Internet

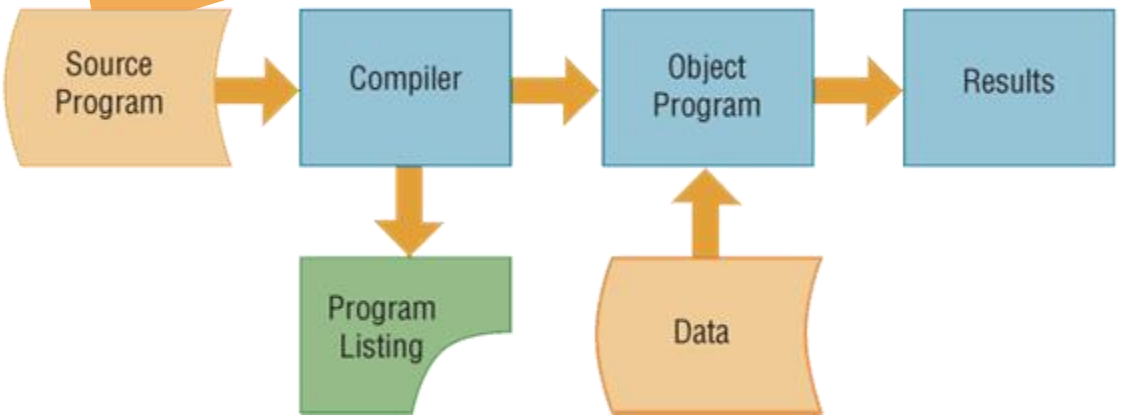
Khủng hoảng phần mềm

- Khái niệm software crisis bao gồm hàng loạt vấn đề nảy sinh trong việc phát triển phần mềm trong hững năm 1960s khi muốn xây dựng những hệ thống phần mềm lớn trên cơ sở các kỹ thuật phát triển thời đó.
- Kết quả:
  1. Thời gian và giá thành tăng vọt tới mức không thể chấp nhận nổi.
  2. Năng xuất không đáp ứng yêu cầu.
  3. Chất lượng phần mềm bị giảm, thấp.
- Để giải quyết các vấn đề kể trên , chuyên ngành software engineering ra đời.

Chương trình biên dịch

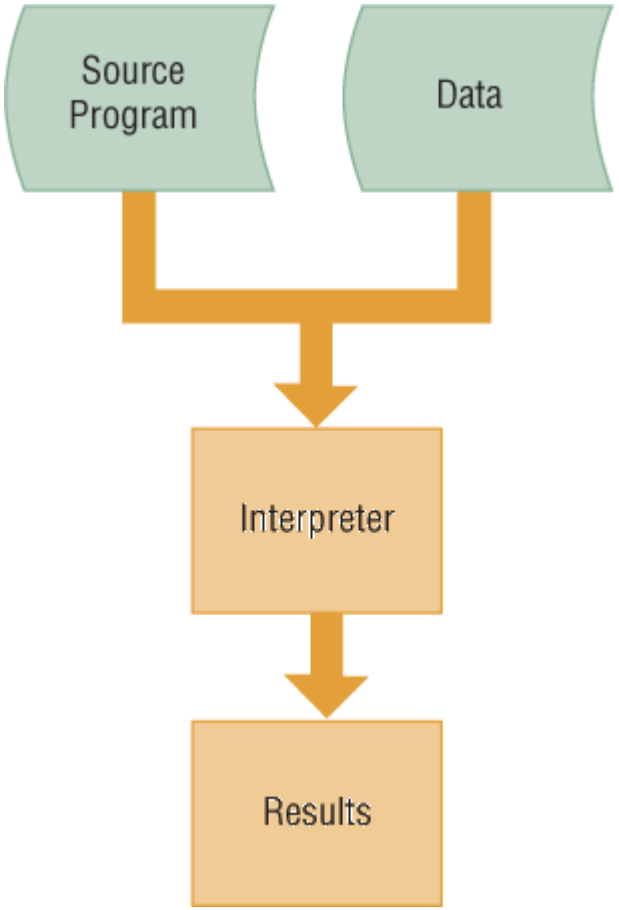
- Biên dịch - Compiler?
  - **Là chương trình thực hiện biên dịch toàn bộ chương trình nguồn thành mã máy trước khi thực hiện**

```
*    COMPUTE REGULAR TIME PAY
     MULTIPLY REGULAR-TIME-HOURS BY HOURLY-PAY-RATE
         GIVING REGULAR-TIME-PAY.

*    COMPUTE OVERTIME PAY
     IF OVERTIME-HOURS > 0
         COMPUTE OVERTIME-PAY = OVERTIME-HOURS * 1.5 * HOURLY-PAY-RATE
     ELSE
         MOVE 0 TO OVERTIME-PAY.

*    COMPUTE GROSS PAY
     ADD REGULAR-TIME-PAY TO OVERTIME-PAY
         GIVING GROSS-PAY.

*    PRINT GROSS PAY
     MOVE GROSS-PAY TO GROSS-PAY-OUT.
     WRITE REPORT-LINE-OUT FROM DETAIL-LINE
         AFTER ADVANCING 2 LINES.
```



Chương trình thông dịch

- Thông dịch - Interpreter?
  - **Là chương trình dịch và thực hiện từng dòng lệnh của chương trình cùng lúc**
  - **Không tạo ra object program**

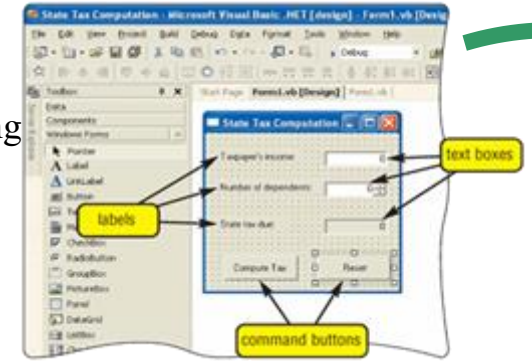


p. 667 Fig. 13-5

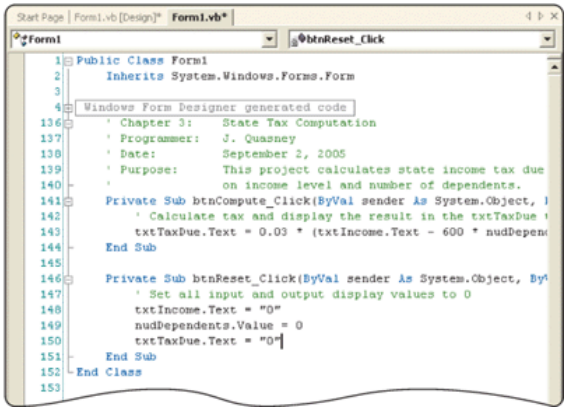
Object-Oriented Programming Languages

- Visual Studio .NET 2003, 2005?
  - **Bước phát triển của visual programming languages và RAD tools**
  - **.NET là tập hợp các công nghệ cho phép program chạy trên Internet**
  - **Visual Basic .NET 2003-5 dùng để xd các ct hướng đối tượng phức tạp**

Step 1. LTV thiết kế giao diện người dùng - user interface.



Step 2. LTV gán các thuộc tính cho mỗi object trên form.



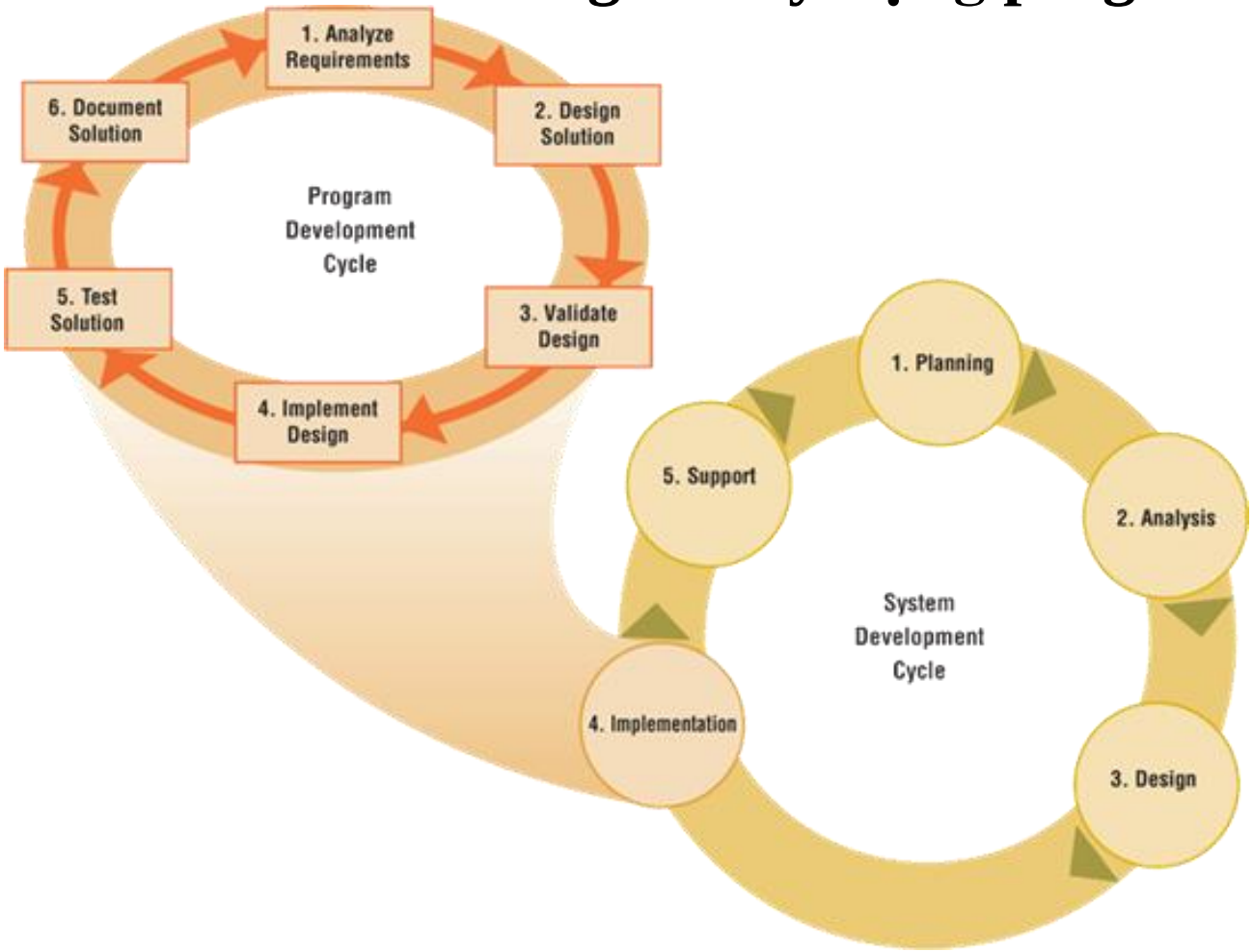
Step 3. LTV viết code để xác định các action cần thực hiện đối với các sự kiện cần thiết.

Step 4. LTV kiểm tra application.

Next

Chu trình phát triển phần mềm

- **Là các bước mà LTV dùng để xây dựng programs**



Step 1 — Analyze Requirements

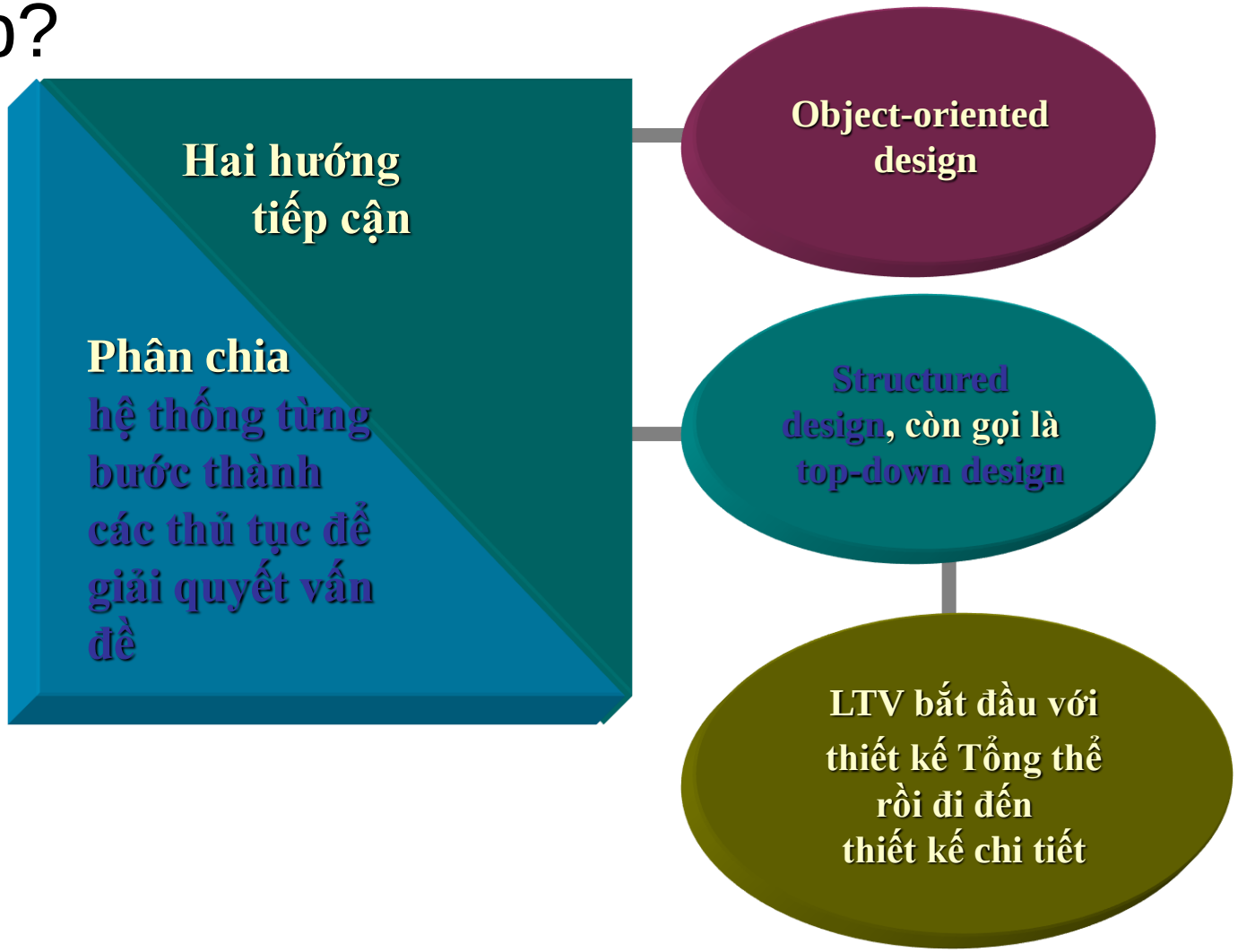
- Các việc cần làm khi phân tích yêu cầu?
  1. **Thiết lập các requirements**
  2. **Gặp các nhà phân tích hệ thống và users**
  3. **Xác định input, output, processing, và các thành phần dữ liệu**
     - **IPO chart**—Xác định đầu vào, đầu ra và các bước xử lý

IPO CHART

Input	Processing	Output
Regular Time Hours Worked	Read regular time hours worked, overtime hours worked, hourly pay rate.	Gross Pay
Overtime Hours Worked	Calculate regular time pay.	
Hourly Pay Rate	If employee worked overtime, calculate overtime pay.	
	Calculate gross pay.	
	Print gross pay.	

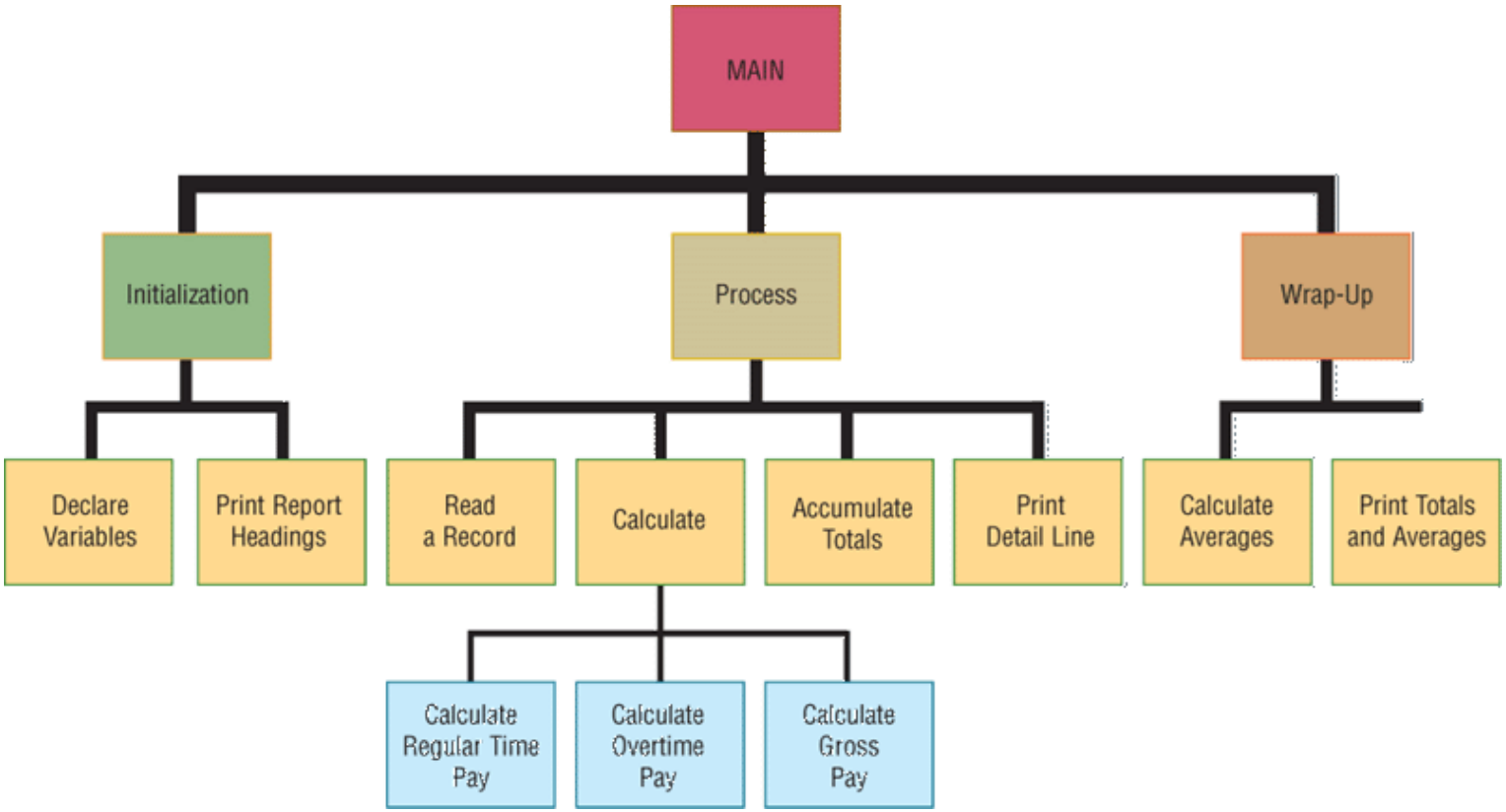
Step 2 — Design Solution

- Những việc cần làm trong bước thiết kế giải pháp?



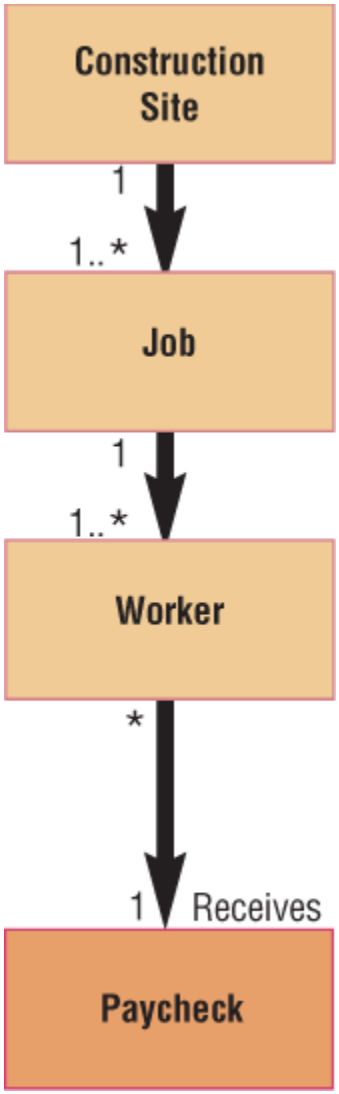
Step 2 — Design Solution

- Sơ đồ phân cấp chức năng- hierarchy chart?
  - **Trực quan hóa các modules**
  - **Sơ đồ cấu trúc**



Step 2 — Design Solution

- Object-oriented (OO) design là gì?
  - **LTV đóng gói dữ liệu và các thủ tục xử lý dữ liệu trong một object**
    - Các objects được nhóm lại thành các classes
    - Biểu đồ lớp thể hiện trực quan các quan hệ phân cấp quan hệ của các classes



Step 3 — Validate Design

- Những điều cần làm trong giai đoạn này?

Kiểm tra
độ chính xác
của chương trình

LTV kiểm tra
logic cho tính đúng đắn
và thử tìm các lỗi logic

Desk check
LTV dùng các dữ liệu
thử nghiệm để kiểm tra
chương trình

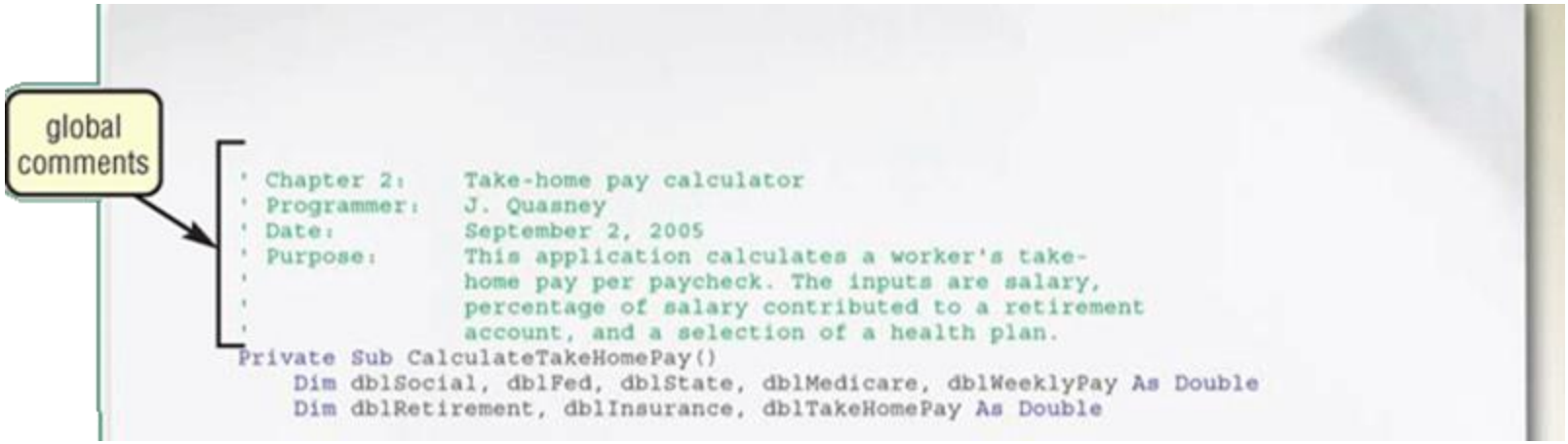
Logic error
các sai sót khi thiết kế
gây ra những kết quả
không chính xác

Test data
các dữ liệu thử nghiệm
giống như số liệu thực mà
chương trình sẽ thực hiện

Structured walkthrough
LTV mô tả logic
của thuật toán trong khi
programming team duyệt theo
logic chương trình

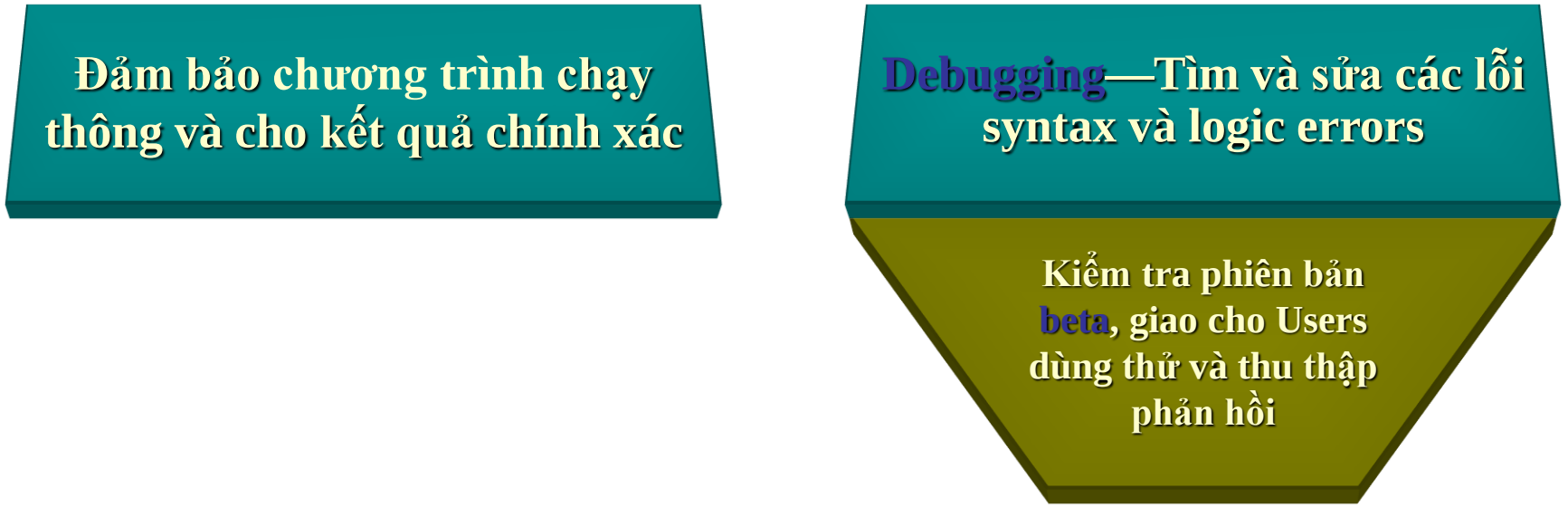
Step 4 — Implementation

- Implementation?
  - **Viết code : dịch từ thiết kế thành program**
    - **Syntax**—Quy tắc xác định cách viết các lệnh
    - **Comments**—program documentation
  - **Extreme programming (XP)—coding và testing ngay sau khi các yêu cầu được xác định**



Step 5 — Test Solution

- Những việc cần làm ?



Step 6 — Document Solution

- Là bước không kém quan trọng
  - **2 hoạt động**

Rà soát lại program code—loại bỏ các **dead code**, tức các lệnh mà **chương trình** không bao giờ gọi đến

Rà soát, hoàn thiện documentation

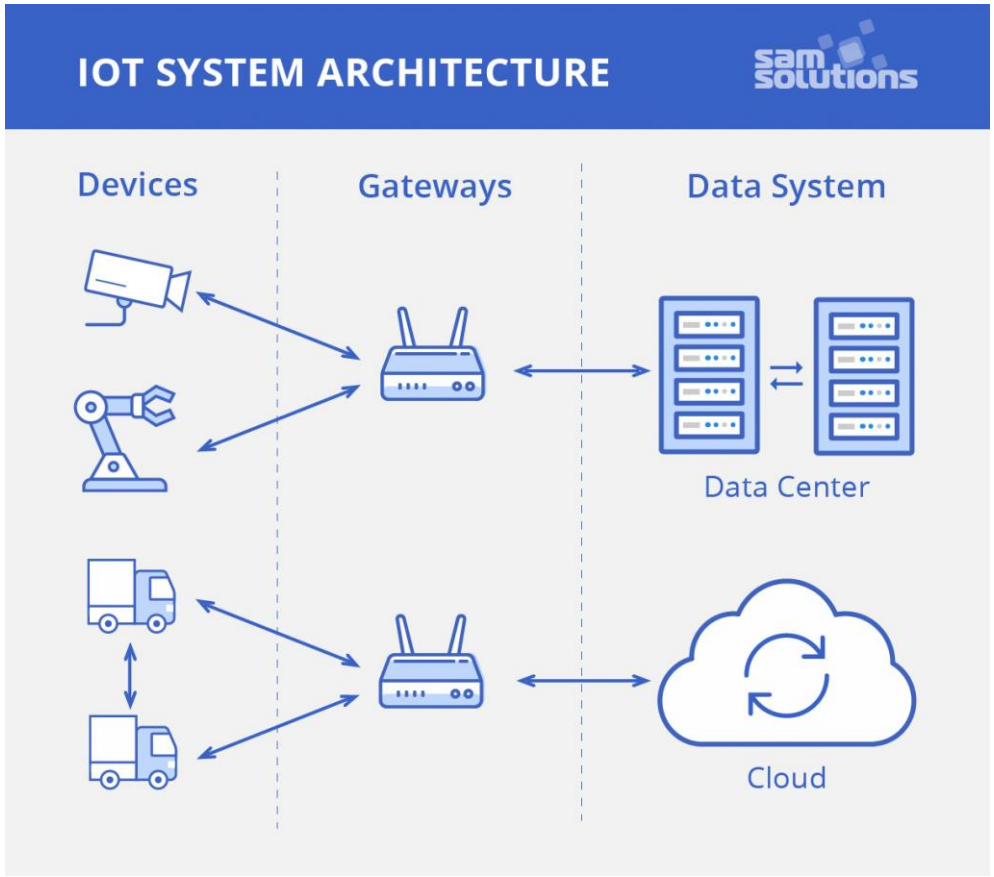
Các mô hình lập trình

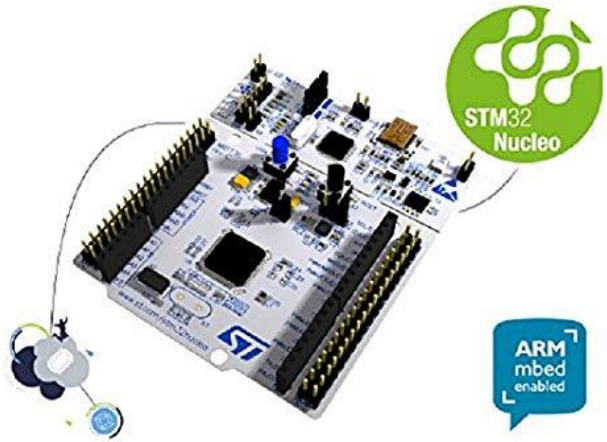
- Programming paradigm
  - · Là một khuôn mẫu - pattern dùng như Mô hình lập trình máy tính
  - . Là một mô hình cho các lớp có cùng những đặc trưng cơ bản
- Programming technique
  - · Liên quan đến các ý tưởng thuật toán để giải quyết một lớp vấn đề tương ứng
  - · Ví dụ: 'Divide and conquer' và 'program development by stepwise refinement'
- ·Programming style
  - . Là cách chúng ta trình bày trong 1 computer program
  - · Phong cách tốt giúp cho chương trình dễ hiểu, dễ đọc, dễ kiểm tra -> dễ bảo trì, cập nhật, gỡ rối, tránh bị lỗi
- · Programming culture
  - · Tổng hợp các hành vi lập trình, thường liên qua đến các dòng ngôn ngữ lập trình
  - · Là tổng thể của Mô hình chính, phong cách và kỹ thuật lập trình
  - . Là nhân cách trong lập trình cũng như khai thác các chương trình

Môn Học Kỹ Thuật Lập Trình

- Cung cấp *kiến thức cơ bản* trong tư duy *lập trình hệ thống* sử dụng ngôn ngữ lập trình C.
- Cung cấp *kiến thức cơ bản* trong tư duy ngôn ngữ lập trình C++.
- Sinh viên sẽ nắm vững được kiến thức về
  - *lập trình có cấu trúc* với các *kiểu dữ liệu có sẵn*
  - Sử dụng C++ như một ngôn ngữ C mở rộng
  - *Lập trinh C++ hướng theo đối tượng (Object oriented design)*
  - *Các cấu trúc dữ liệu điển hình dùng trong hệ thống nhúng (Embedded system)*

Đối tượng lập trình





Tại Sao Học C ?

- C là ngôn ngữ gần với ngôn ngữ máy nhất vì thế được sử dụng tại hầu hết các *hệ thống nhúng*
  - Ngôn ngữ lập trình cấp cao cho hệ vi xử lý: STM32, DsPIC, Atmega,
  - Ngôn ngữ lai: MATLAB, NI CVI, Arduino ?
  - Ngôn ngữ lập trình cho Windows: Visual C++,
- Hệ thống thư viện phần mềm có sẵn trên Internet

  https://github.com/

  www.thecodeproject.com

Không Chỉ C

- Môn học đề cập đến những vấn đề cơ bản nhất của một ngôn ngữ lập trình
  - Biến, kiểu dữ liệu, giá trị, biểu thức
  - Thực thi chương trình tuần tự
- Các cấu trúc dữ liệu và giải thuật thông dụng.
- Và một số khái niệm liên quan đến *thiết kế chương trình*
  - Sự trừu tượng hóa thủ tục, hàm, dữ liệu
  - Sự đóng gói, module hóa, tính sử dụng lại

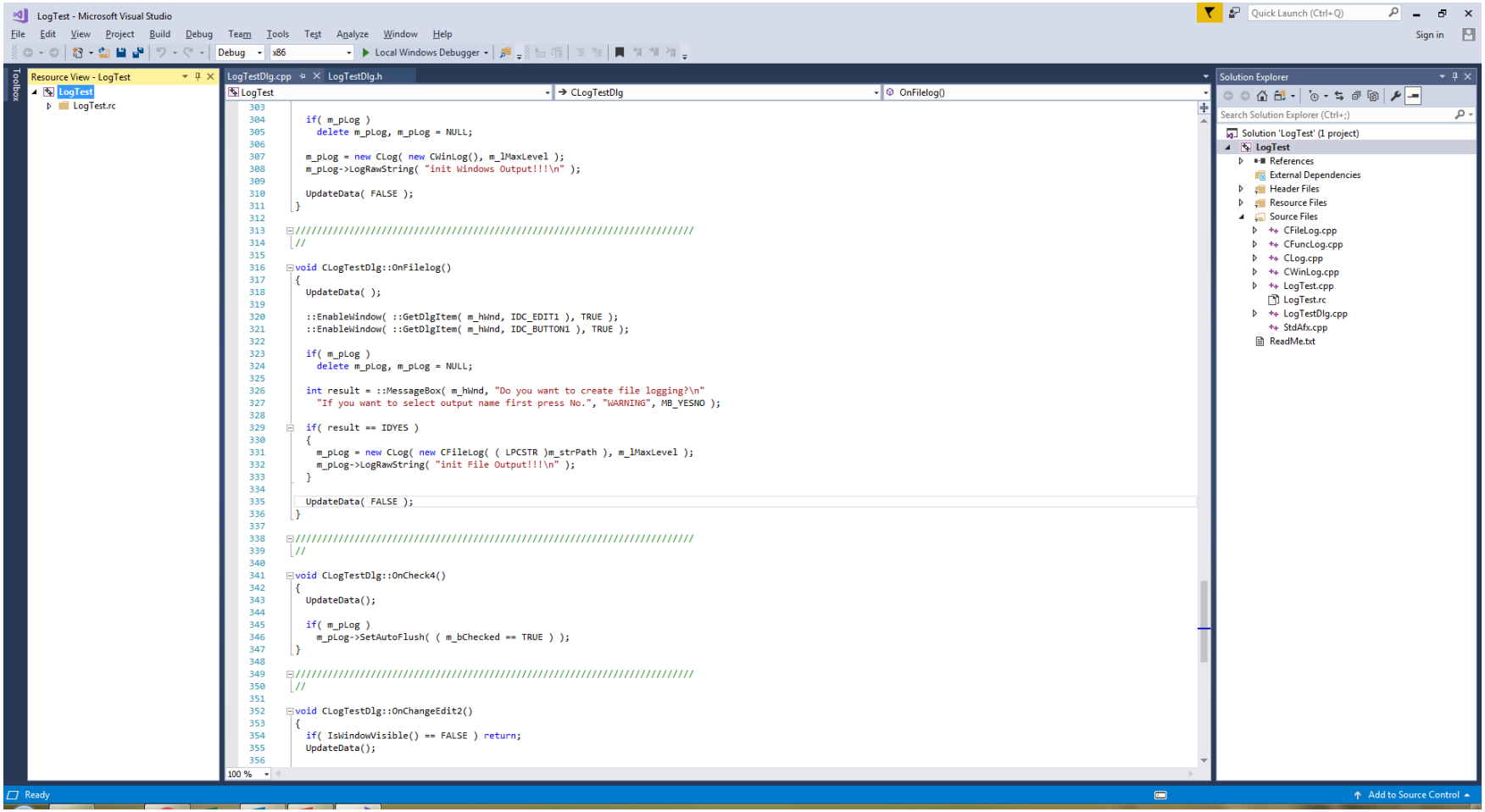
Giải Quyết Vấn Đề / Thiết Kế Chương Trình

- *Xác định* rõ vấn đề.
- *Phân tích* vấn đề.
- Thiết kế *thuật toán* nhằm giải quyết vấn đề.
- *Cài đặt* thuật toán (viết chương trình).

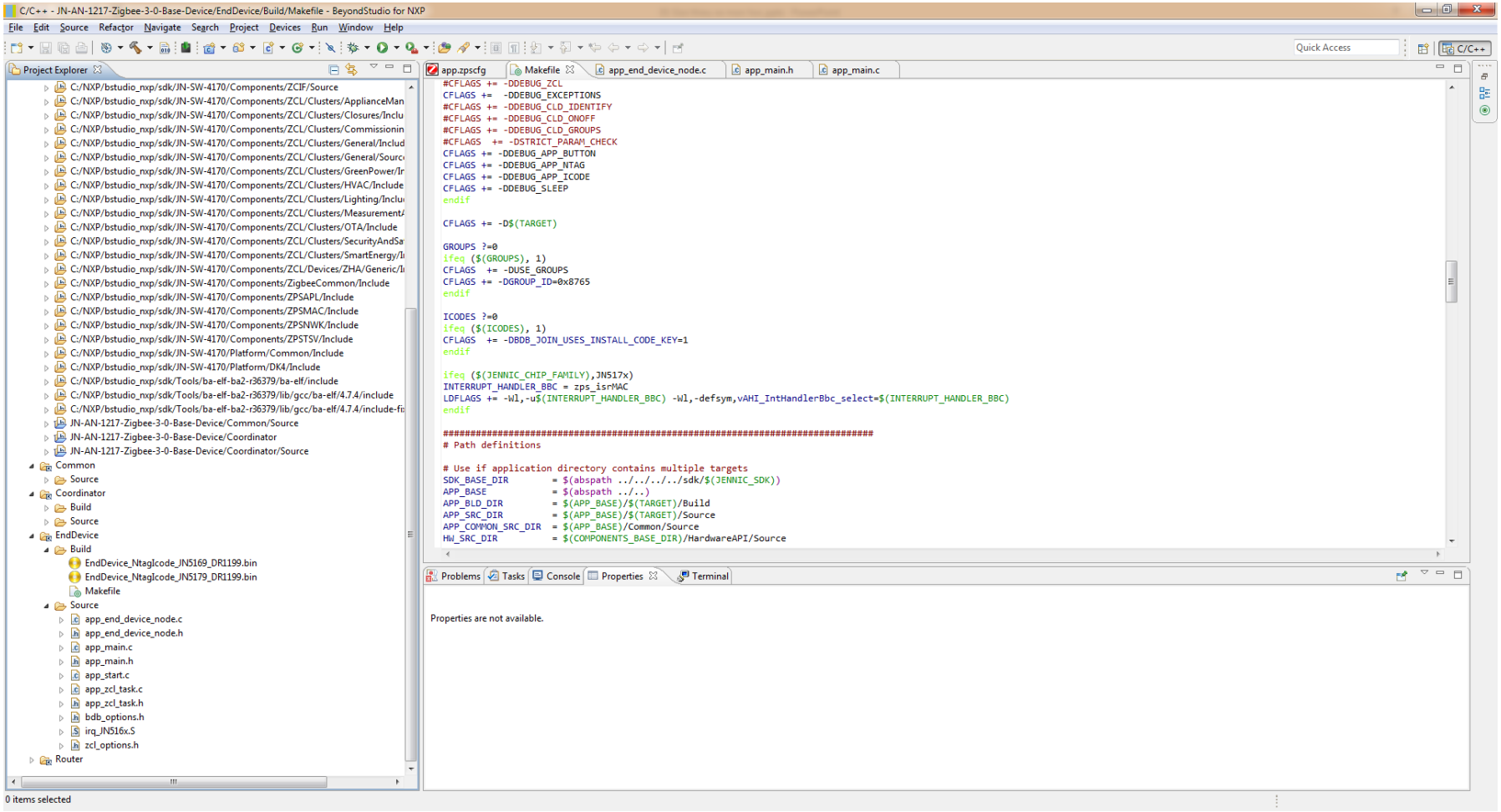
  (nhớ ghi chú thích trong chương trình)
- *Chạy thử* và xác nhận tính đúng đắn của chương trình.

  (chu trình chạy thử - sửa lỗi)
- *Duy trì* và cập nhật chương trình.

Những Gì Đang Chờ Đợi Bạn tương lai



Những Gì Đang Chờ Đợi Bạn tương lai



Các yêu cầu ngắn hạn

- Điểm số: trải đều từ 0 đến 10 với trung bình từ 6 đến 7.
- Một môn học khó với nội dung đề cập đến nhiều vấn đề.
- Bài tập ngắn về nhà hàng tuần, một bài tập dài cho cả kỳ, bài kiểm tra ngắn trên lớp hàng tuần.
- Bài kiểm tra giữa kỳ và cuối kỳ trong 1 tiếng với nội dung
  - Câu hỏi lý thuyết
  - Kiểm tra kết quả chương trình
  - Viết chương trình giải quyết vấn đề thực tế
- Vậy có gì thú vị?

Lời Khuyên Với Sinh Viên

- Học nghiêm túc ngay từ đầu kỳ.
- Có thói quen tìm kiếm sự trợ giúp sớm và thường xuyên đối với những nội dung khó.
- Hoàn thành, chạy thử và gỡ lỗi các bài tập ngắn của từng tuần trước ở nhà vì chúng thường liên quan đến bài kiểm tra ngắn kế tiếp.
- Nộp bài tập dài đúng hạn (bài tập nộp muộn sẽ nhận điểm 0, không nộp bài tập sẽ nhận điểm âm).
- Có thể đặt câu hỏi bất cứ lúc nào.

Tài Liệu Tham Khảo

- *Kernighan B.W., Ritchie D.M., The C Programming Language, 2nd edition, Prentice Hall, 1998*
- *Kernighan B.W., Pike R., The Practice of Programming, Addison Wesley, 1999*
- *Trần Việt Linh, Lê Đăng Hưng, Lê Đức Trung, Nguyễn Thanh Thủy, Nhập Môn Lập Trình Ngôn Ngữ C, NXB Khoa Học & Kỹ Thuật, 2003*
- *Microsoft Developer Network (MSDN)*
- *Internet*

Liên Hệ

- Cán bộ phụ trách môn học
  - **Nguyễn Hồng Quang -** quang.nguyenhong1@hust.edu.vn
  - **Nguyễn Trí Cường -** cuong.nguyentri@hust.edu.vn
- Bộ môn Tự động hóa CN (C9-104), Viện Điện,

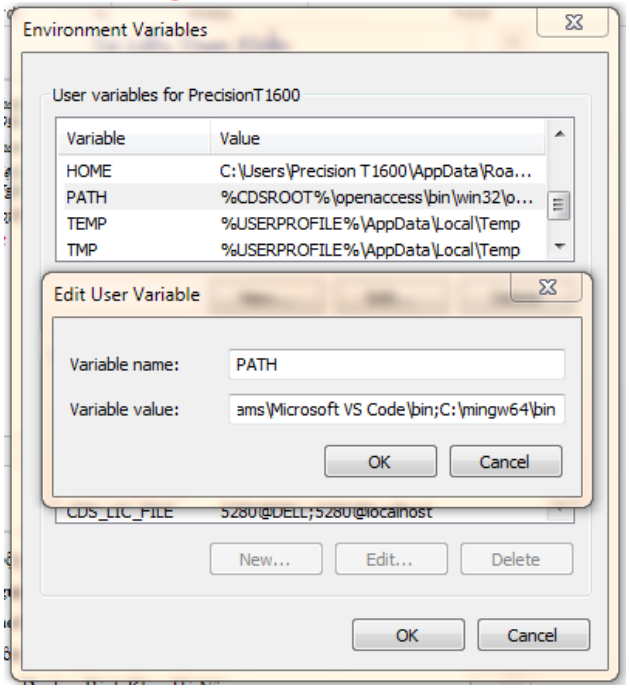
  Trường Đại học Bách Khoa Hà Nội.

Hãy Bắt Đầu!

- Cài đặt phần mềm trên nền hệ điều hành Windows

https://code.visualstudio.com/docs/cpp/config-mingw

  - Cài đặt Visual Studio Code.
  - Cài đặt C++ extension for VSCode
  - Cài đặt Mingw-w64, tại thư mục đơn giản như C:\Mingw64
  - Cài đặt đường dẫn path cho file g++.exe



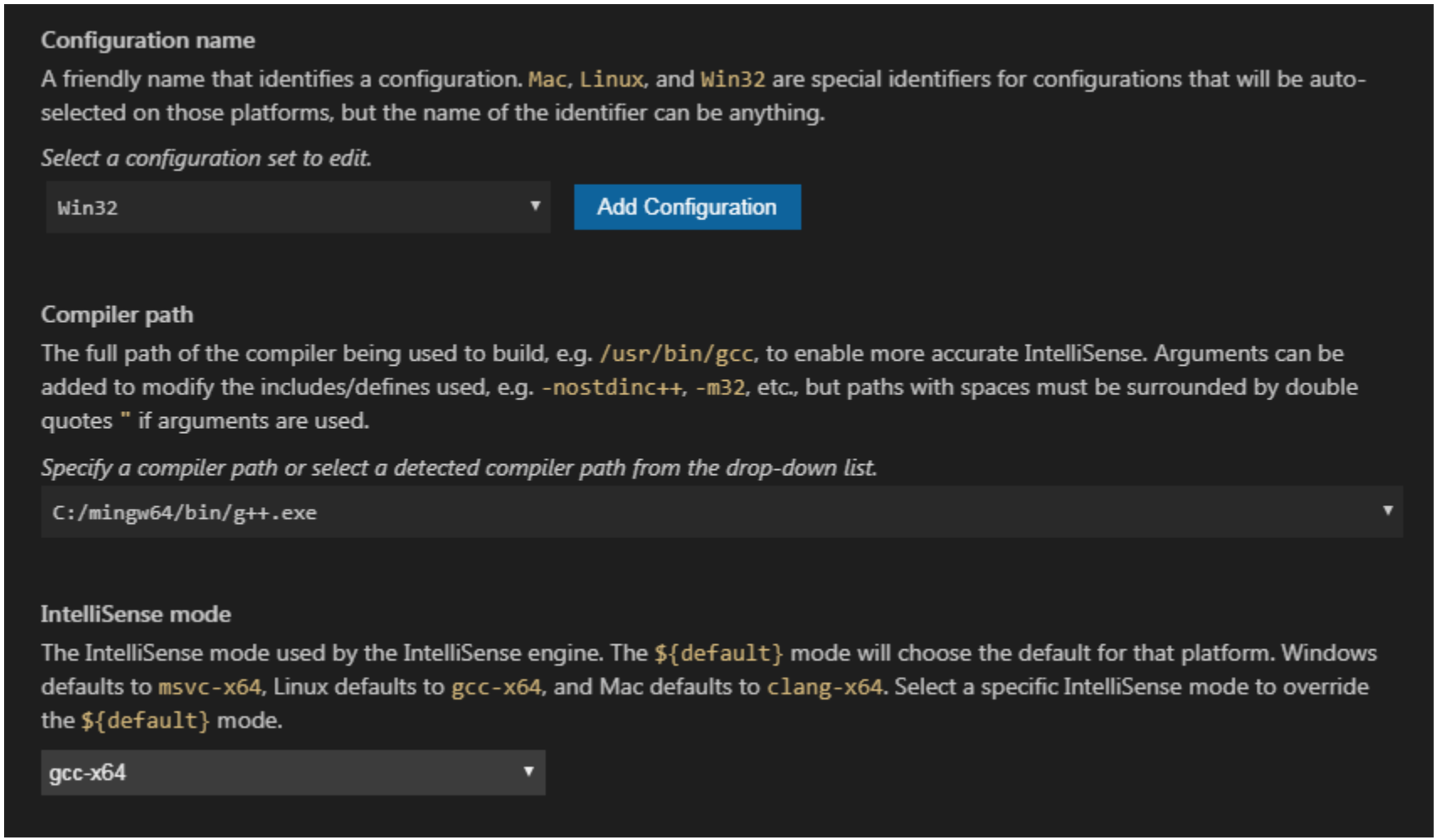
Các file cấu hình đi kèm

- *c_cpp_properties.json* (compiler path and IntelliSense settings)
- *tasks.json* (build instructions)
- *launch.json* (debugger settings)

(.json JavaScript Object Notation (JSON) format)

Đường dẫn cho trình biên dịch

(compiler path)



Tạo tác vụ dịch

(build task)

```
{
    "version": "2.0.0",
    "tasks": [
        {
            "label": "build hello world",
            "type": "shell",
            "command": "g++",
            "args": [
                "-g",
                "-o",
                "hello",
                "hello.cpp"
            ],
            "group": {
                "kind": "build",
                "isDefault": true
            }
        },
        {
            "type": "shell",
            "label": "g++.exe build active file",
            "command": "C:\\mingw64\\bin\\g++.exe",
            "args": [
                "-g",
                "${file}",
                "-o",
                "${fileDirname}\\${fileBasenameNoExtension}.exe"
            ],
```

Cấu hình chế độ debug

(launch.json)

```
{
    // Use IntelliSense to learn about possible attributes.
    // Hover to view descriptions of existing attributes.
    // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
    "version": "0.2.0",
    "configurations": [
        {
            "name": "(gdb) Launch",
            "type": "cppdbg",
            "request": "launch",
            "program": "${workspaceFolder}/hello.exe",
            "args": [],
            "stopAtEntry": true,
            "cwd": "${workspaceFolder}",
            "environment": [],
            "externalConsole": true,
            "MIMode": "gdb",
            "miDebuggerPath": "C:\\mingw64\\bin\\gdb.exe",
            "setupCommands": [
                {
                    "description": "Enable pretty-printing for gdb",
                    "text": "-enable-pretty-printing",
                    "ignoreFailures": true
                }
            ]
        }
    ]
}
```