

# **IT 3040 - KỸ THUẬT LẬP TRÌNH**

## **Năm học 2018-2019**

**Giảng viên: PGS. TS.Huỳnh Quyết Thắng**

**BM Công nghệ phần mềm**

**Viện CNTT-TT, ĐHBK HN**

**<https://users.soict.hust.edu.vn/thangq>**

**Điện thoại: 0913536752**

# **Chương 1. Những khái niệm cơ bản trong kỹ thuật lập trình**

---

1. Tổng quan về lập trình
2. Bài toán lập trình
3. Quy trình giải quyết bài toán lập trình
4. Ngôn ngữ lập trình và các thách thức công nghệ

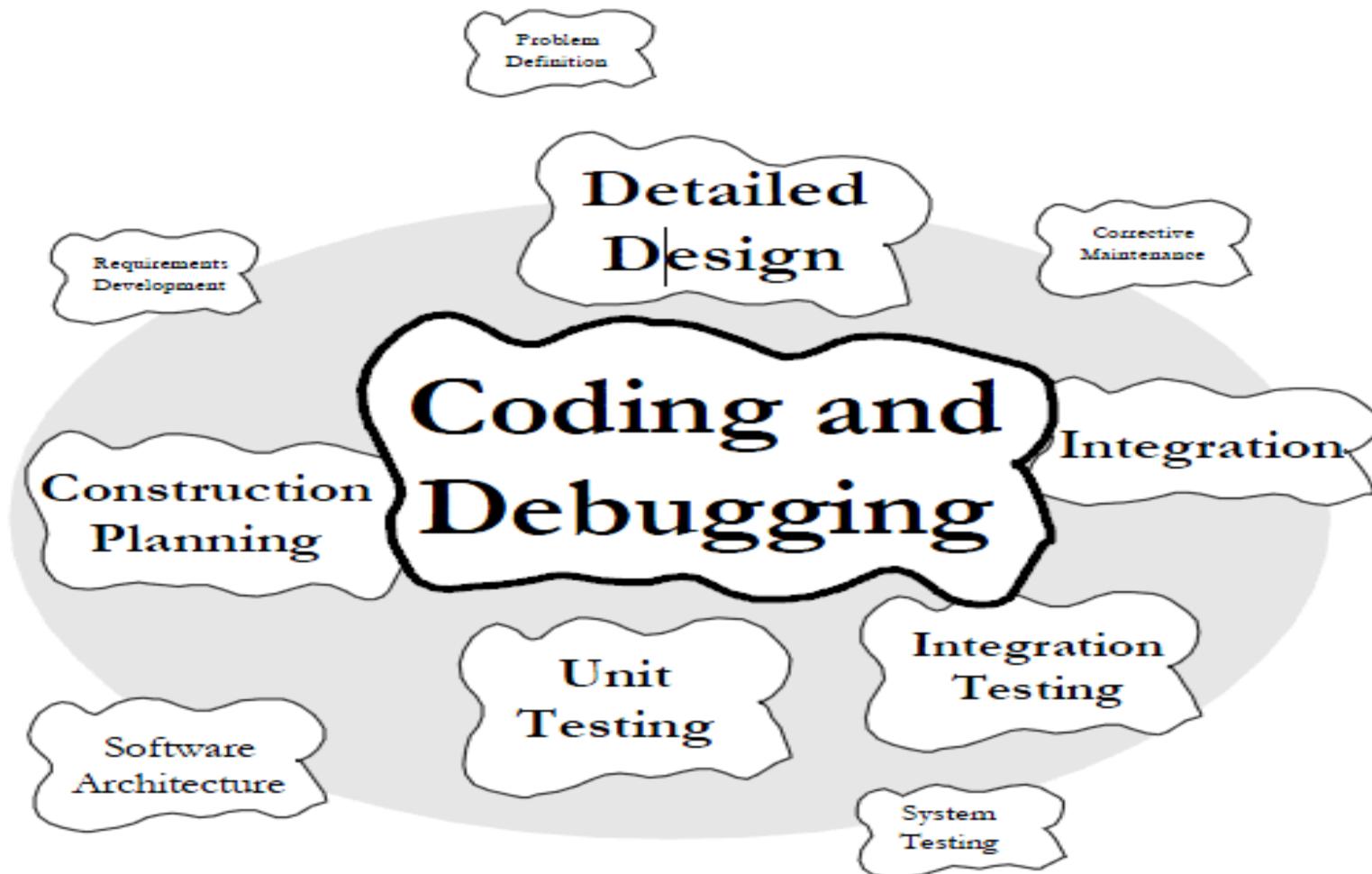
# Tổng quan về lập trình

---

Software Construction 11 nội dung cơ bản:

- Định nghĩa bài toán/Problem definition
- Xác định yêu cầu/Requirements development
- Kế hoạch xây dựng/Construction planning
- Kiến trúc phần mềm / Software architecture, or high-level design
- Thiết kế chi tiết/ Detailed design
- Lập trình và gỡ lỗi/Coding and debugging
- Kiểm thử thành phần/Unit testing
- Kiểm thử tích hợp/Integration testing
- Tích hợp phần mềm/Integration
- Kiểm thử hệ thống/System testing
- Bảo trì/Corrective maintenance

# Tổng quan về lập trình



# Tổng quan về lập trình

---

## Lập trình – những kỹ thuật cơ bản

- Creating and naming variables and named constants
- Selecting control structures and organizing blocks of statements
- Unit testing, integration testing, and debugging your own code
- Reviewing other team members' low-level designs and code and having them review yours
- Polishing code by carefully formatting and commenting it
- Integrating software components that were created separately
- Tuning code to make it faster and use fewer resources

## Tổng quan về lập trình

---

Tại sao lập trình lại quan trọng:

Construction is a large part of software development

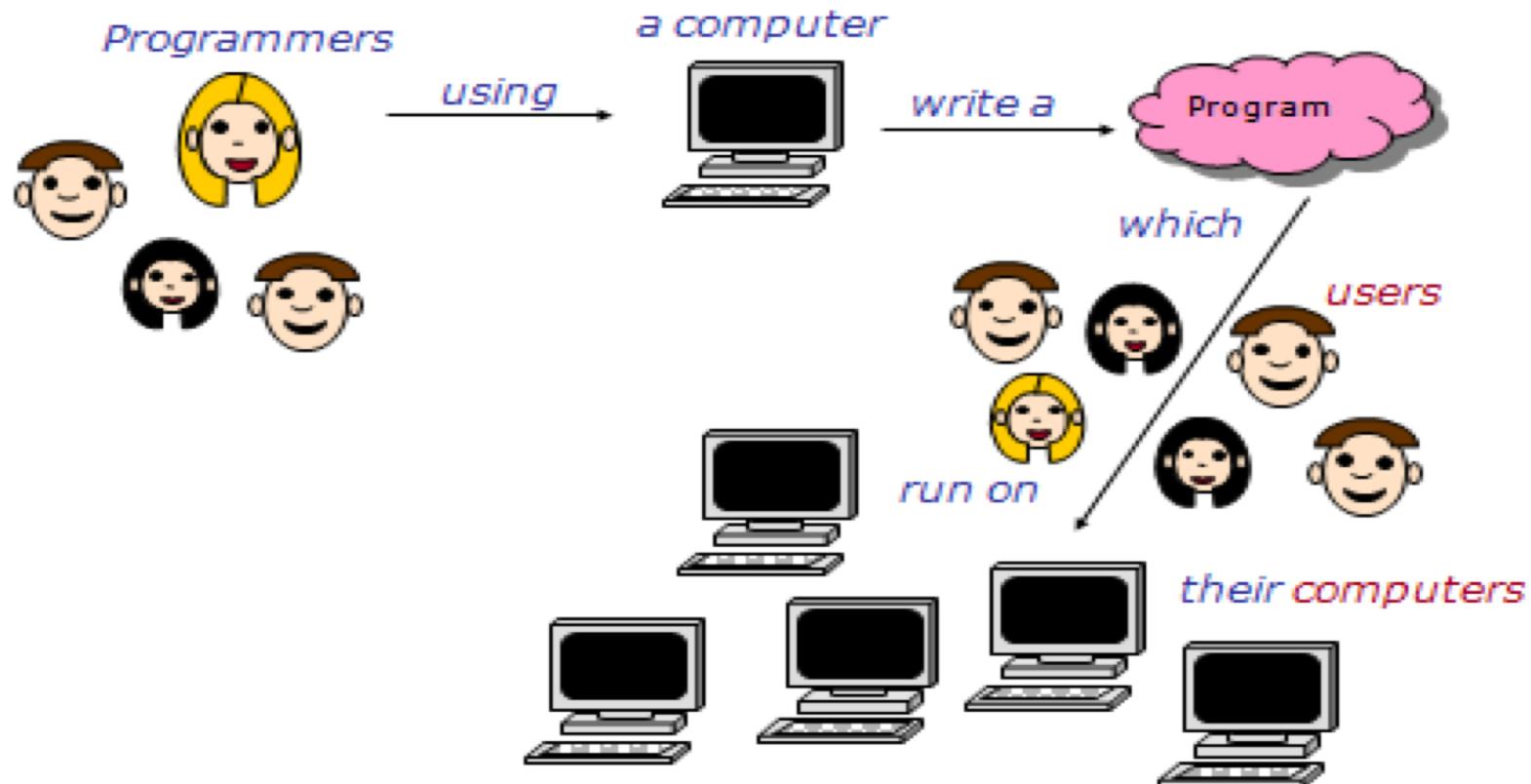
Construction is the central activity in software development

Construction's product, the source code, is often the only  
accurate description of the software

Construction is the only activity that's guaranteed to be done

# Bài toán lập trình

## Working with a computer



# Bài toán lập trình

## Computers



Computers are universal machines. They execute the program that you feed them.



+

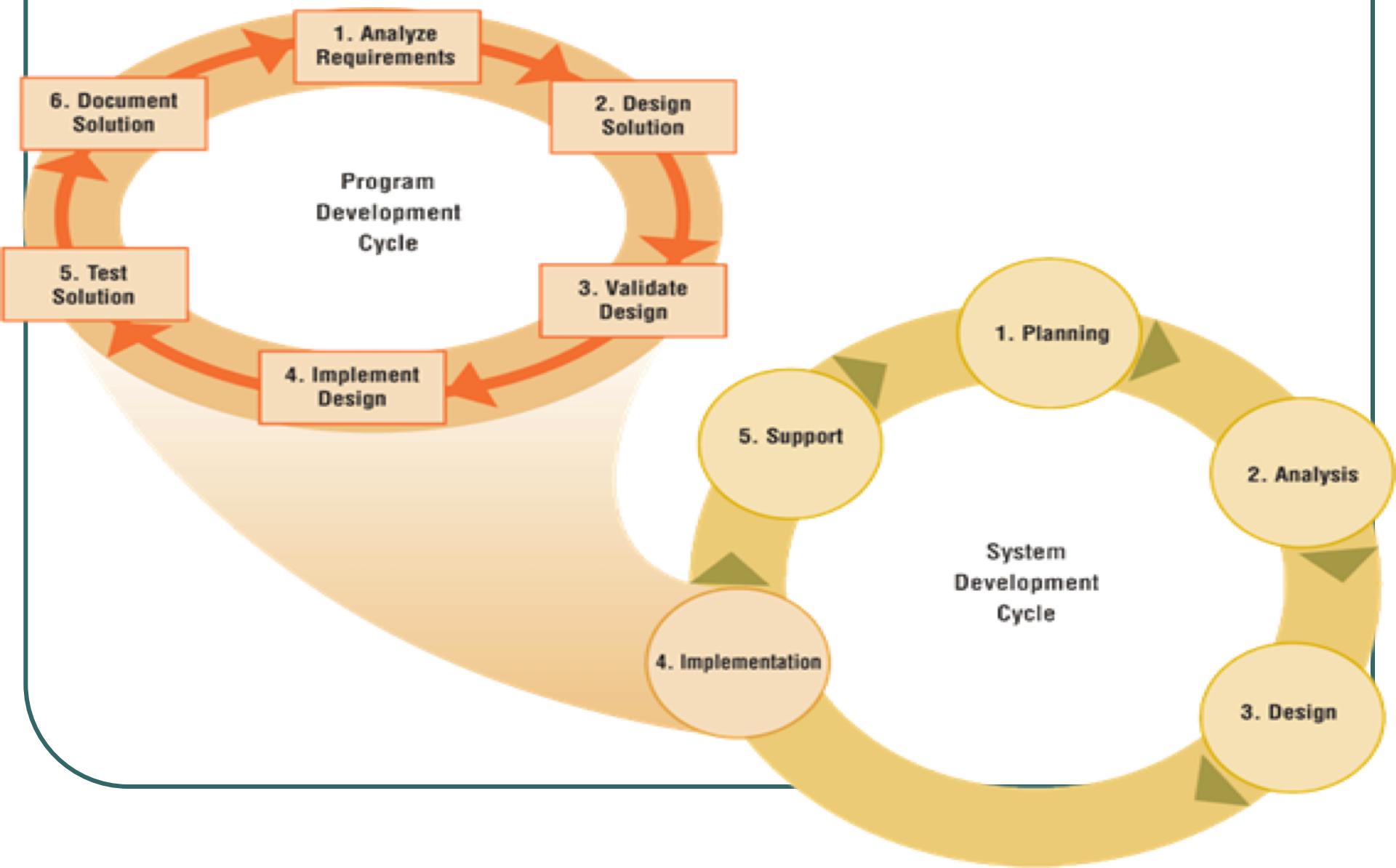
Program

=

Specialized  
machine

(Universal  
machine)

# Quy trình giải quyết bài toán lập trình



# 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.	

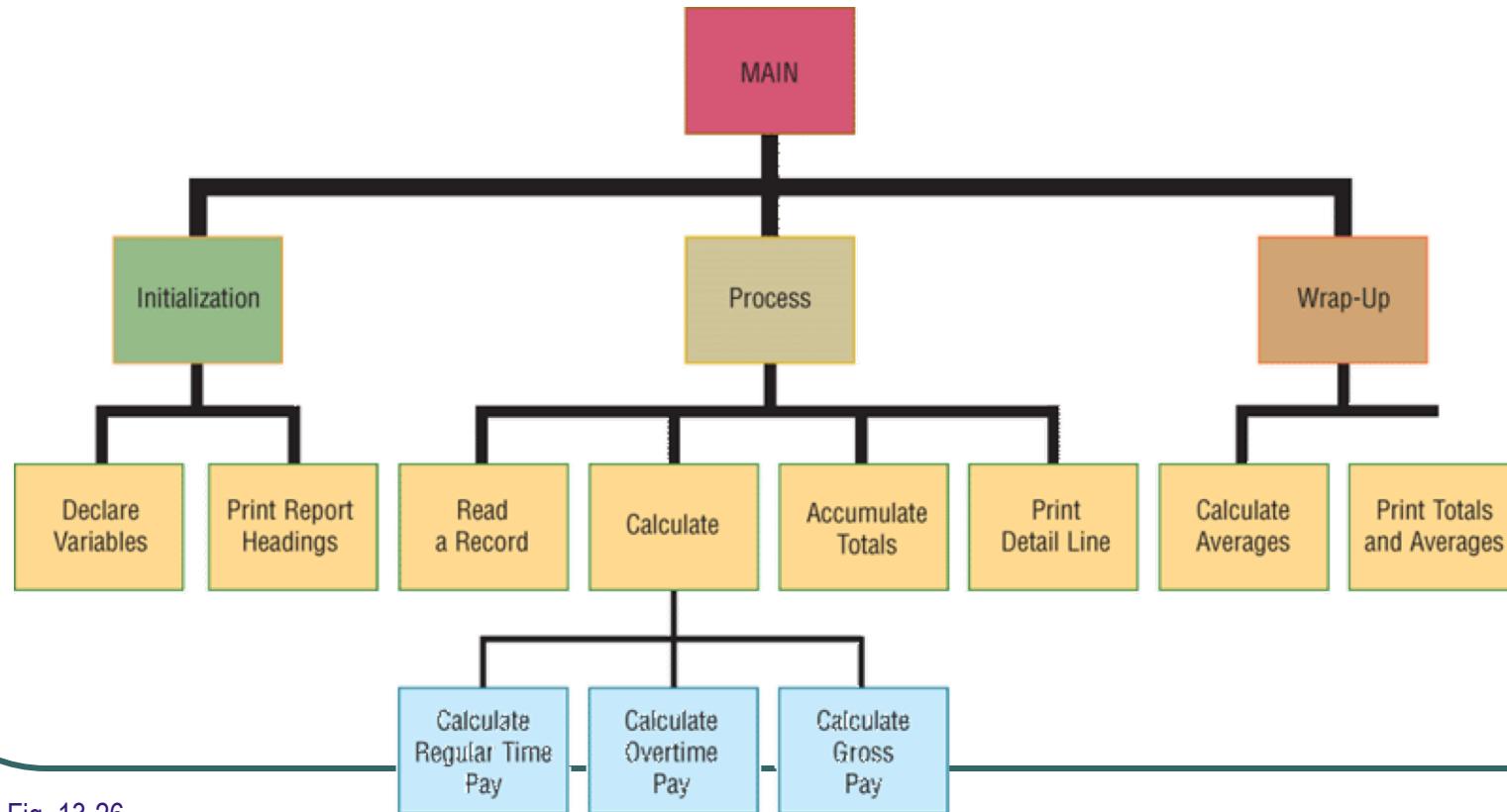
## Step 2 — Design Solution

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



## Step 2 — Design Solution

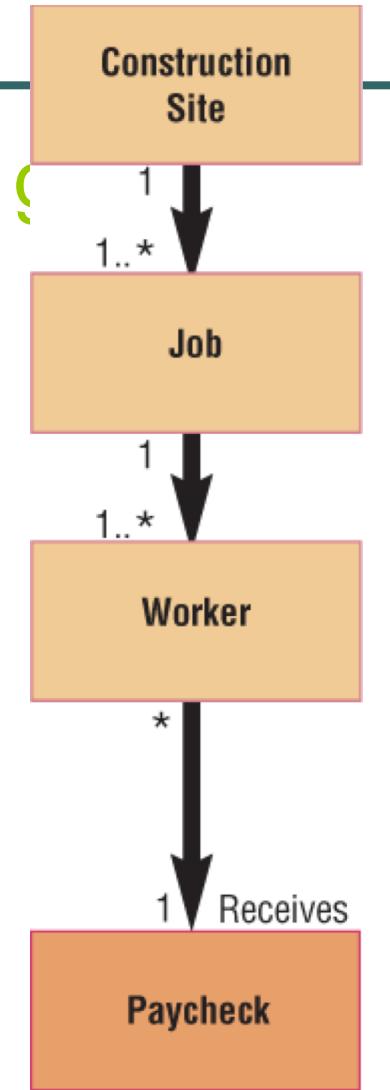
- Sơ đồ phân cấp chức năng- hierarchy
- Trục quan hóa các modules ct
- Còn gọi là sơ đồ cấu trúc



## Step 2 — Design Solution

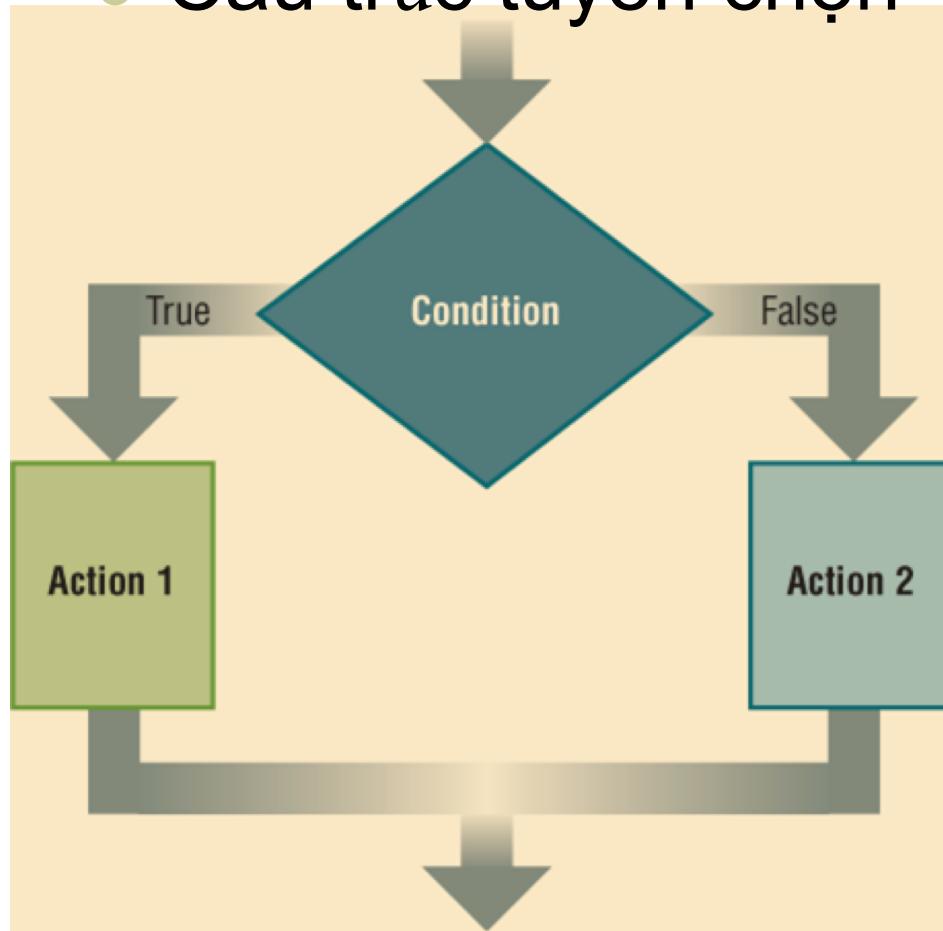
- Object-oriented (OO) design là cách LTV đóng gói dữ liệu và các thủ tục xử lý dữ liệu trong 1 object

- 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 2 — Design Solution

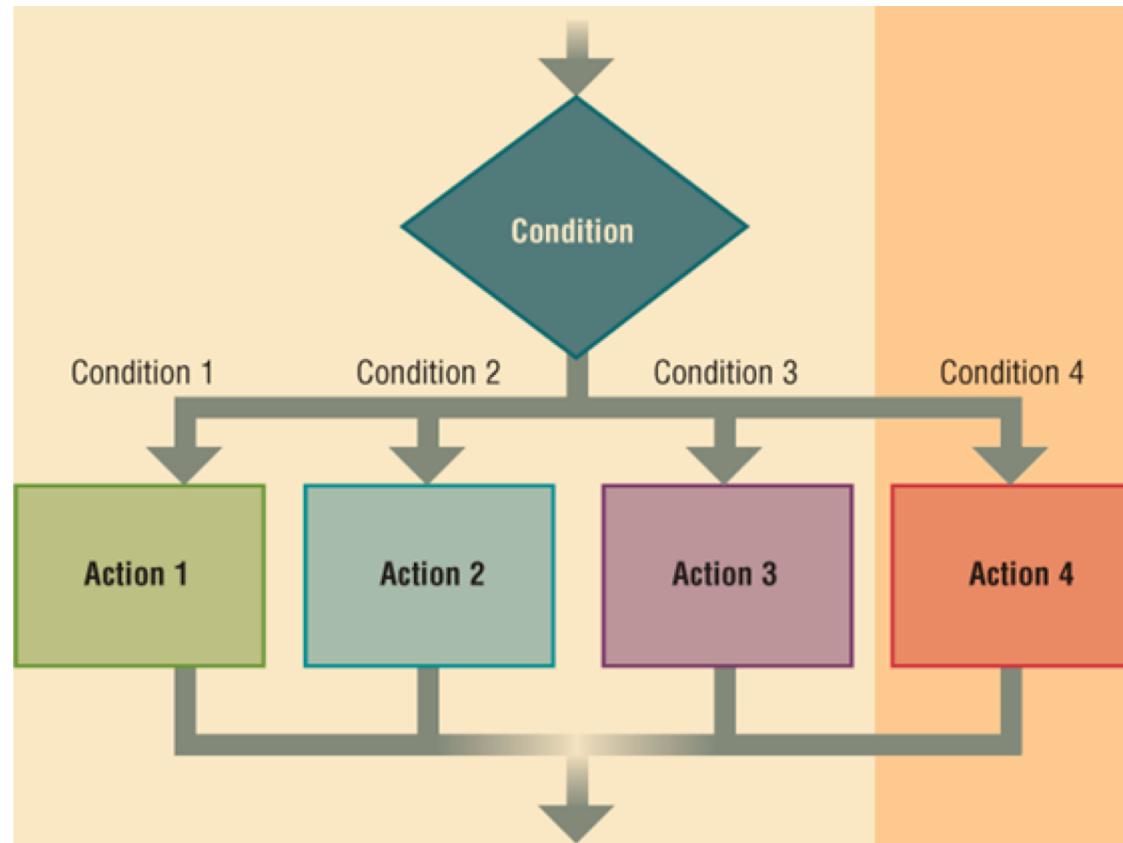
- Cấu trúc tuyển chọn



- Chỉ ra action tương ứng điều kiện 2 kiểu
  - Case control structure
  - **If-then-else control structure**—dựa theo 2 khả năng: true or false

## Step 2 — Design Solution

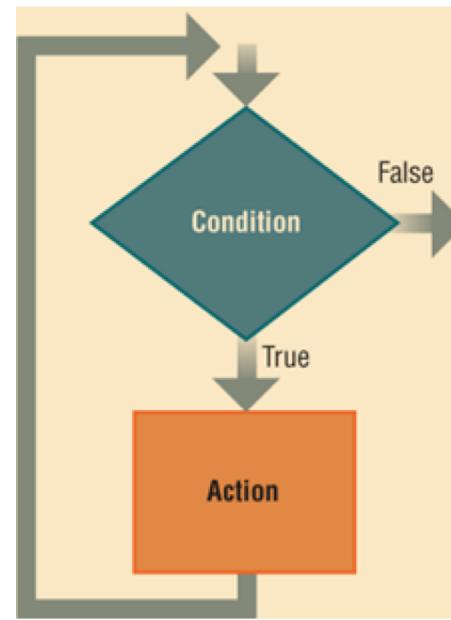
- Case control structure
  - Dựa theo 3 hoặc nhiều hơn các khả năng



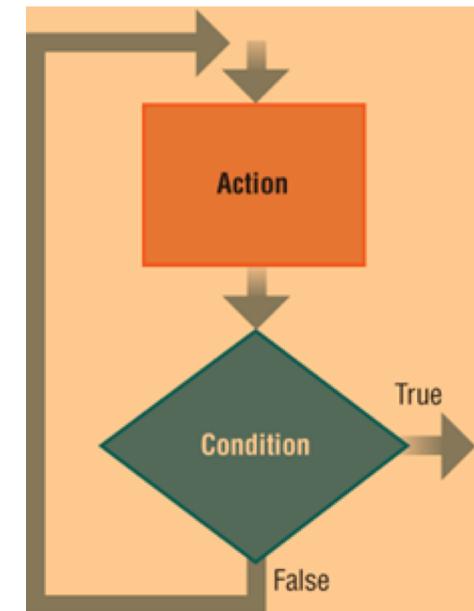
## Step 2 — Design Solution

- Cấu trúc lặp  
➤ Cho phép CT thực hiện 1 hay nhiều actions lặp đi lặp lại

- Do-while control structure—lặp khi điều kiện còn đúng
- Do-until control structure—Lặp cho đến khi điều kiện đúng



Do-While Control Structure



Do-Until Control Structure

# Step 3 — Validate Design

## ● Những điều cần làm trong giai đoạn này?

Kiểm tra

độ chính xác  
của program

Desk check

LTV dùng các dữ liệu  
thử nghiệm để kiểm tra ct

Test data

các dữ liệu thử nghiệm  
giống như số liệu thực mà  
CT sẽ thực hiện

LTV kiểm tra

logic cho tính đúng đắn  
và thử tìm các lỗi logic

Logic error

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

Structured walkthrough

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

# Step 4 — Implement Design

## ● 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**

```
' Chapter 2:      Take-home pay calculator
' Programmer:    J. Quasney
' Date:          September 2, 2005
' Purpose:        This application calculates a worker's take-
'                 home pay per paycheck. The inputs are salary,
'                 percentage of salary contributed to a retirement
'                 account, and a selection of a health plan.
Private Sub CalculateTakeHomePay()
    Dim dblSocial, dblFed, dblState, dblMedicare, dblWeeklyPay As Double
    Dim dblRetirement, dblInsurance, dblTakeHomePay As Double
```

# Step 5 — Test Solution

## ● Những việc cần làm ?

Đảm bảo CT chạy thông và cho kq chính xác

Debugging—Tìm và sửa các lỗi syntax và logic errors

Kiểm tra phiên bản beta, giao cho Users dùng thử và thu thập phản hồi

## 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 ct không bao giờ gọi đến

Rà soát, hoàn thiện documentation

# **Chương 1. Những khái niệm cơ bản trong kỹ thuật lập trình**

---

1. Tổng quan về lập trình
2. Bài toán lập trình
3. Quy trình giải quyết bài toán lập trình
4. Ngôn ngữ lập trình
5. Các thách thức công nghệ

# CÁC THẾ HỆ NNLT

## LANGUAGE GENERATIONS

• Generation	Classification
1st	Machine languages
2nd	Assembly languages
3rd	Procedural languages
4th	Application languages (4GLs)
5th	AI techniques, inference languages
6th	Neural networks (?), others....

## ● Low-level languages và high-level languages?

### Low-level language

### High-level language

Machine-dependent

Phụ thuộc phần cứng, chỉ chạy trên  
một loại máy tính

Machine-independent

Thường không phụ thuộc phần  
cứng, có thể chạy trên nhiều loại  
máy tính khác nhau

Machine và assembly languages là  
ngôn ngữ bậc thấp low-level

## PHÂN LOẠI THEO MỨC ĐỘ TRÙU TƯỢNG

• Level	Instructions	Memory handling
Low level languages	Dạng bits – giống các lệnh machine	Truy cập và cấp phát trực tiếp bộ nhớ
High level languages	Dùng các biểu thức và các dòng điều khiển xác định	Truy cập và cấp phát bộ nhớ qua các lệnh, toán tử - operators
Very high level languages	Hoàn toàn trùu tượng, độc lập phần cứng	Che dấu hoàn toàn việc truy cập và tự động cấp phát bộ nhớ

## DECLARATIVE và NON-DECLARATIVE PROGRAMMING

---

- Các ngôn ngữ có thể chia thành 2 nhóm : “Cái gì cần lưu trữ” và “Lưu trữ như thế nào”.
- Nhóm 1 gọi là **declarative** (tường thuật -chính là functional và logic languages).
- Nhóm 2 gọi là **non-declarative** hay **procedural** (tức là các ngôn ngữ mệnh lệnh).

# Procedural Languages – Ngôn ngữ thủ tục

## ● Procedural language?

Lập trình viên viết các chỉ thị hướng dẫn cho máy tính cai gì cần làm và làm như thế nào

Sử dụng hàng loạt các từ giống tiếng anh để viết các chỉ thị - instructions

Còn gọi là third-generation language (3GL)

Các ngôn ngữ thông dụng là BASIC, COBOL, PASCAL, C,C++ và JAVA



Click to view animation

# Procedural Languages

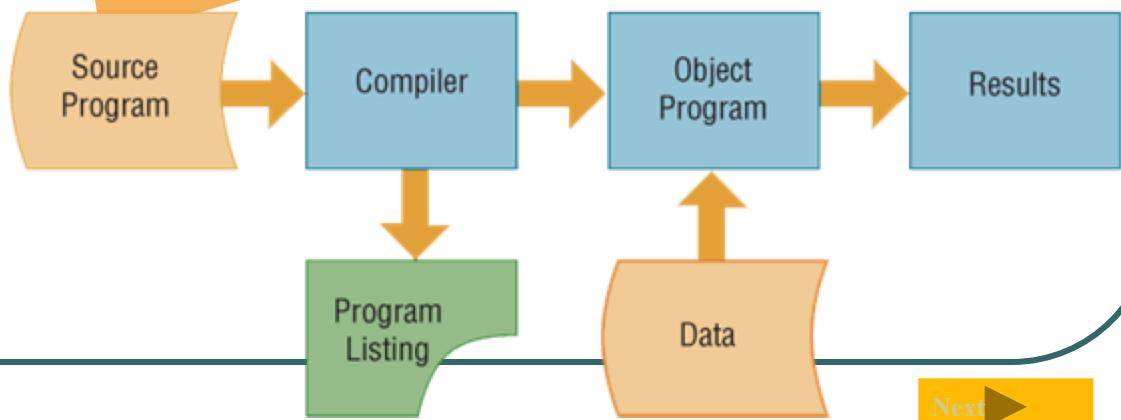
- Trình dịch - Compiler  
Là chương trình thực hiện biên dịch toàn bộ chương trình nguồn thành mã máy trước khi thực hiện

```
* 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.

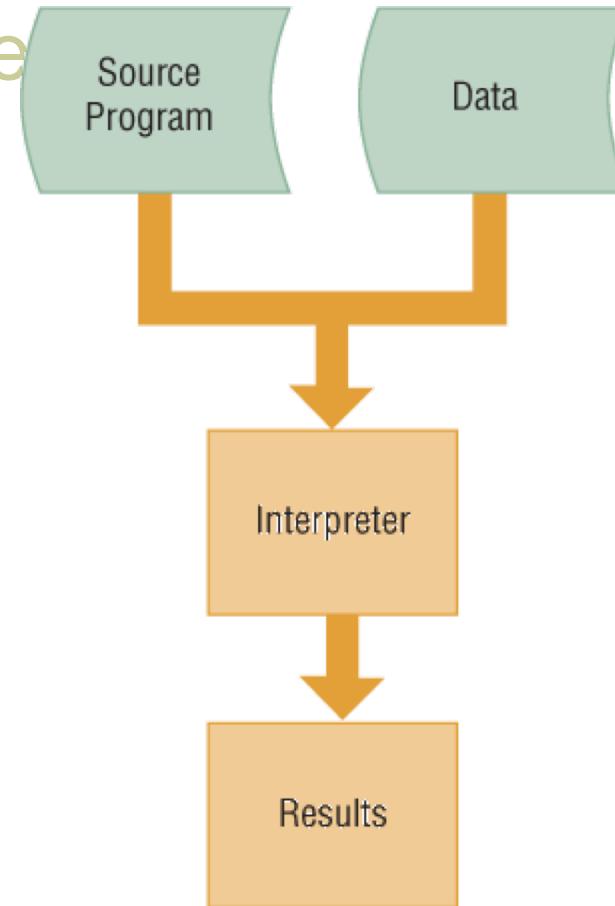
* COMPUTE GROSS PAY
ADD REGULAR-TIME-PAY TO OVERTIME-PAY
GIVING GROSS-PAY.

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



# Procedural Languages

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



# Procedural Languages

- **BASIC?**
- Được thiết kế để cho những người mới học tiếp cận một cách đơn giản NNLT
- Beginner's All-purpose Symbolic Instruction Code

```
REM COMPUTE REGULAR TIME PAY
Regular.Time.Pay = Regular.Time.Hours * Hourly.Pay.Rate

REM COMPUTE OVERTIME PAY
If Overtime.Hours > 0 THEN
    Overtime.Pay = Overtime.Hours * 1.5 * Hourly.Pay.Rate
ELSE
    Overtime.Pay = 0
END IF

REM COMPUTE GROSS PAY
Gross.Pay = Regular.Time.Pay + Overtime.Pay
REM PRINT GROSS PAY
PRINT USING "The gross pay is $##,###.##"; Gross.Pay
```

# Procedural Languages

## ● COBOL?

- Dùng cho các ứng dụng trong kinh tế
- Các lệnh giống tiếng anh làm cho code dễ đọc, viết và chỉnh sửa
- Common Business-Oriented Language

```
* 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.
```

# Procedural Languages

## ● C?

- Là NNLT rất mạnh, ban đầu được thiết kế để lập trình hệ thống -write system software
- Yêu cầu những kỹ năng lập trình chuyên nghiệp

```
/* Compute Regular Time Pay */  
rt_pay = rt_hrs * pay_rate;  
  
/* Compute Overtime Pay */  
if (ot_hrs > 0)  
    ot_pay = ot_hrs * 1.5 * pay_rate;  
else  
    ot_pay = 0;  
  
/* Compute Gross Pay */  
gross = rt_pay + ot_pay;  
  
/* Print Gross Pay */  
printf("The gross pay is %d\n", gross);
```

# Object-Oriented Programming Languages

## • Object-oriented programming (OOP) language?

Dùng để hỗ trợ thiết kế HĐT object-oriented design

Object là phần tử chứa đựng cả dữ liệu và các thủ tục xử lý dữ liệu

Lợi ích cơ bản là khả năng tái sử dụng -reuse existing objects

Event-driven—  
Hướng sự kiện  
Kiểm tra để trả lời một tập các sự kiện

Event là hành động mà chương trình cần đáp ứng

C++ và Java là các NN hoàn toàn HĐT object-oriented languages

# Object-Oriented Programming Languages

---

- **C++?**

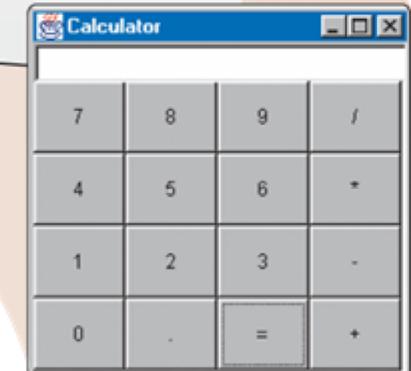
- Chưa đựng các thành phần của C, loại bỏ những nhược điểm và thêm vào những tính năng mới để làm việc với object-oriented concepts
- Được dùng để phát triển các Database và các ứng dụng Web

# Object-Oriented Programming Languages

- **Java?**
- Phát triển bởi Sun Microsystems
- Giống C++ nhưng dùng trình dịch just-in-time (JIT) để chuyển source code thành machine code

```
public void actionPerformed(ActionEvent e)
{
    foundKey = false;

    //Search for the key pressed
    for (int i = 0; i < keysArray.length && !foundKey; i++)
        if(e.getSource() == keysArray[i])      //key match found
    {
        foundKey = true;
        switch(i)
        {
            case 0: case 1: case 2: case 3: case 4: //number buttons
            case 5: case 6: case 7: case 8: case 9: //0 - 9
            case 15:                                //decimal point button
                if(clearText)
                {
                    lcdField.setText("");
                    clearText = false;
                }
                lcdField.setText(lcdField.getText() + keysArray[i].getLabel());
                break;
        }
    }
}
```



# Object-Oriented Programming Languages

## ● Visual programming language?

Visual programming environment (VPE)  
Cho phép developers kéo và thả các objects để xd programs

Cung cấp giao diện trực quan hoặc đồ họa để tạo source code

Đôi khi được gọi là fifth-generation language

Thường được dùng trong môi trường RAD (rapid application development)

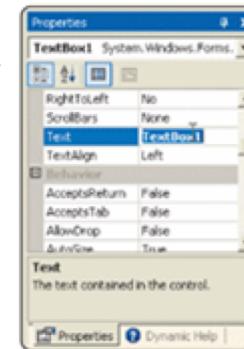
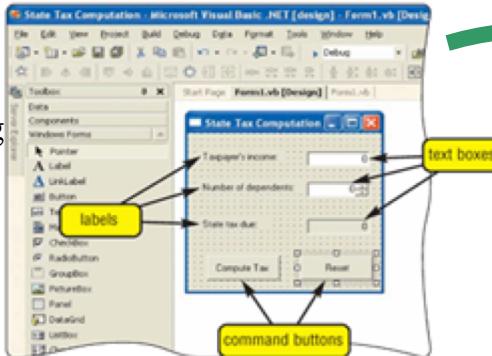
LTV viết và phát triển chương trình trong các segments

# 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.

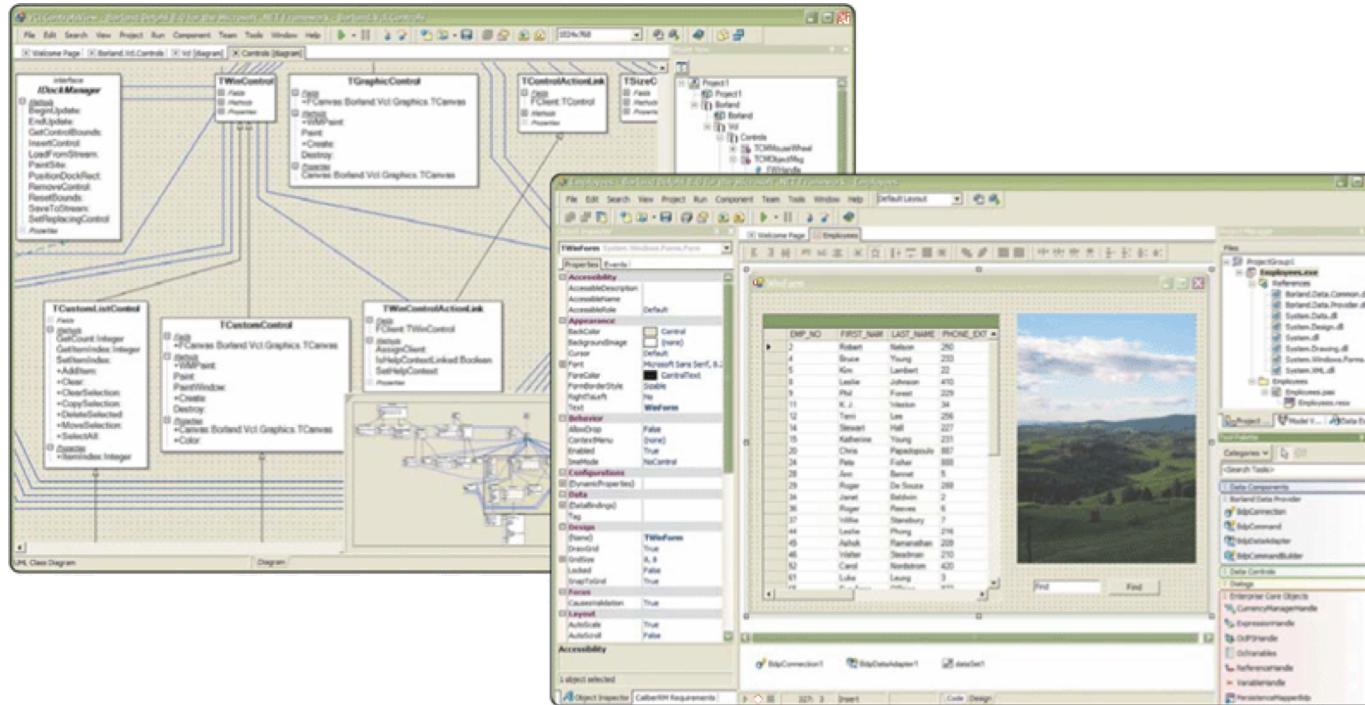
**Step 4.** LTV kiểm tra  
application.

Next ➤

# Object-Oriented Programming Languages

## ● Delphi?

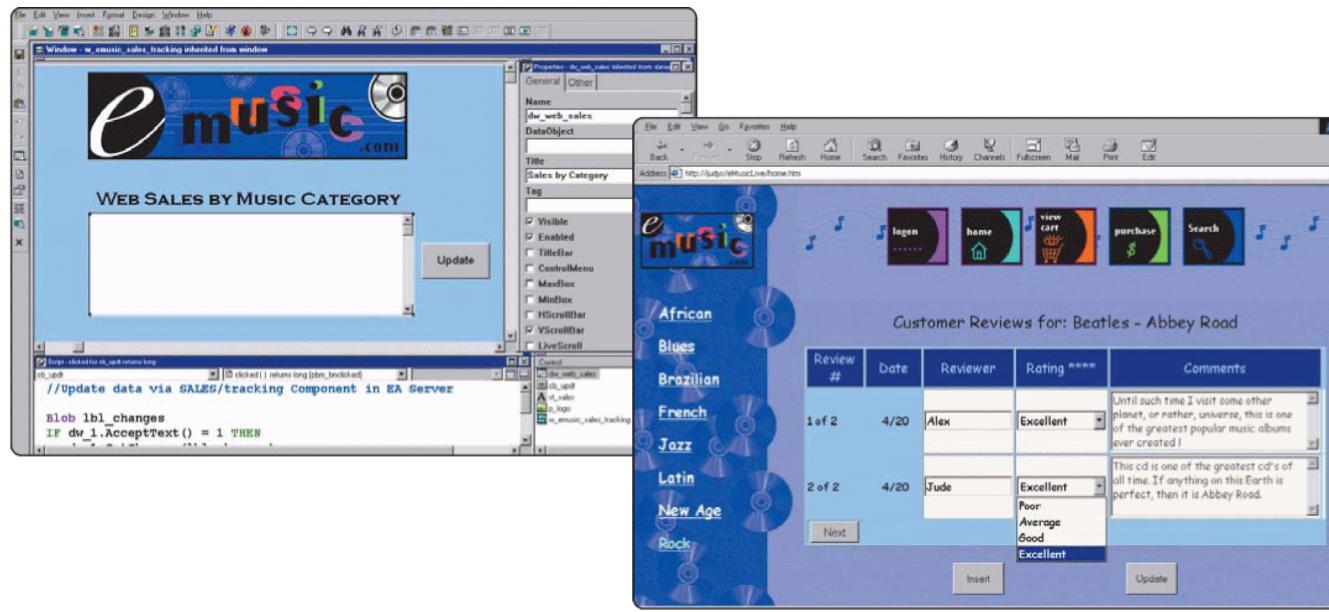
- Là 1 công cụ lập trình trực quát mạnh
- Hợp với những ứng dụng chuyên nghiệp và Web lớn



# Object-Oriented Programming Languages

## ● PowerBuilder?

- Một công cụ lập trình trực quan mạnh khác
- Phù hợp với các ứng dụng Web-based hay các ứng dụng lớn HĐT - object-oriented applications



## Nonprocedural Languages and Program Development Tools

- nonprocedural languages và program development tools?

### Nonprocedural Language

LTV viết các lệnh giống tiếng anh hoặc tương tác với môi trường trực quan để nhận được các dữ liệu từ files hay database

### Program Development Tools

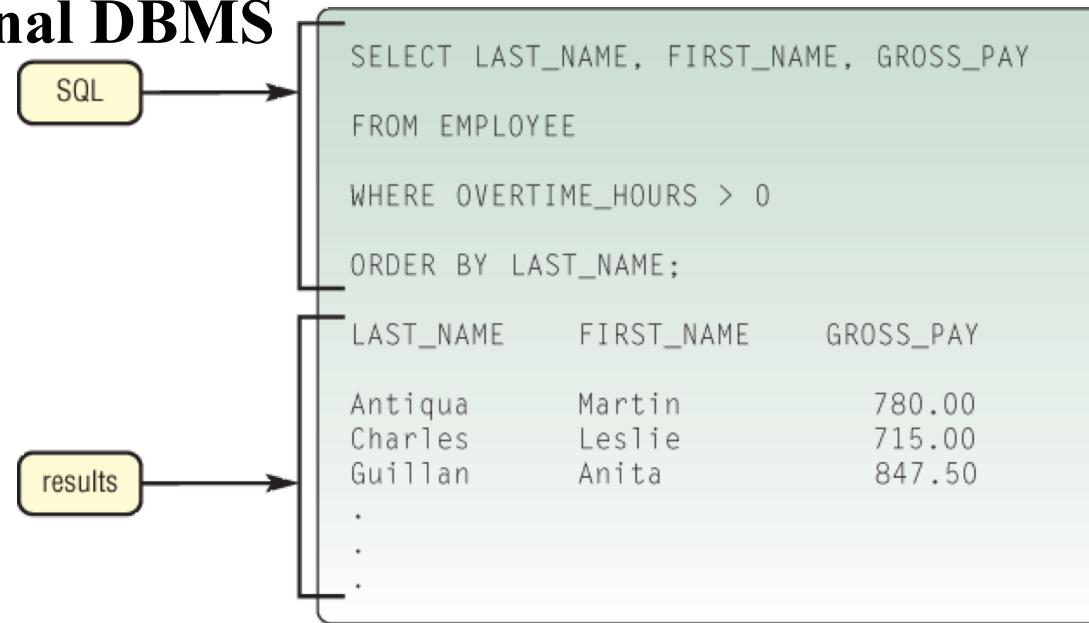
Các chương trình thân thiện với người sử dụng được thiết kế để trợ giúp cả LTV lẫn người sử dụng trong việc tạo chương trình

- **RPG (Report Program Generator)?**
- Các ngôn ngữ LT phi thủ tục được dùng để tạo các báo cáo, thiết lập các thao tác tính toán và cập nhật files

```
C* COMPUTE REGULAR TIME PAY
C      RTHRS      MULT RATE          RTPAY      72
C*
C* COMPUTE OVERTIME PAY
C      OTHRS      IFGT 0           OTRATE      72
C      RATE       MULT 1.5          OTPAY      72
C      OTRATE     MULT OTHRS
C                  ELSE
C                  INZ            OTPAY      72
C
C* COMPUTE GROSS PAY
C      RTPAY      ADD  OTPAY          GRPAY      72
C
C* PRINT GROSS PAY
C                  EXCPTDETAIL
C
C*
O* OUTPUT SPECIFICATIONS
OOPRINT E      DETAIL
O                  23 'THE GROSS PAY IS $'
O      GRPAY J      34
```

# Nonprocedural Languages and Program Development Tools

- NN thế hệ IV fourth-generation language
- Là các ngôn ngữ phi thủ tục cho phép truy cập dữ liệu trong csdl (4GL)?
- NNLT 4GL thông dụng là SQL ,Access, là các ngôn ngữ truy vấn . Cho phép users quản trị dữ liệu trong csdl quan hệ relational DBMS



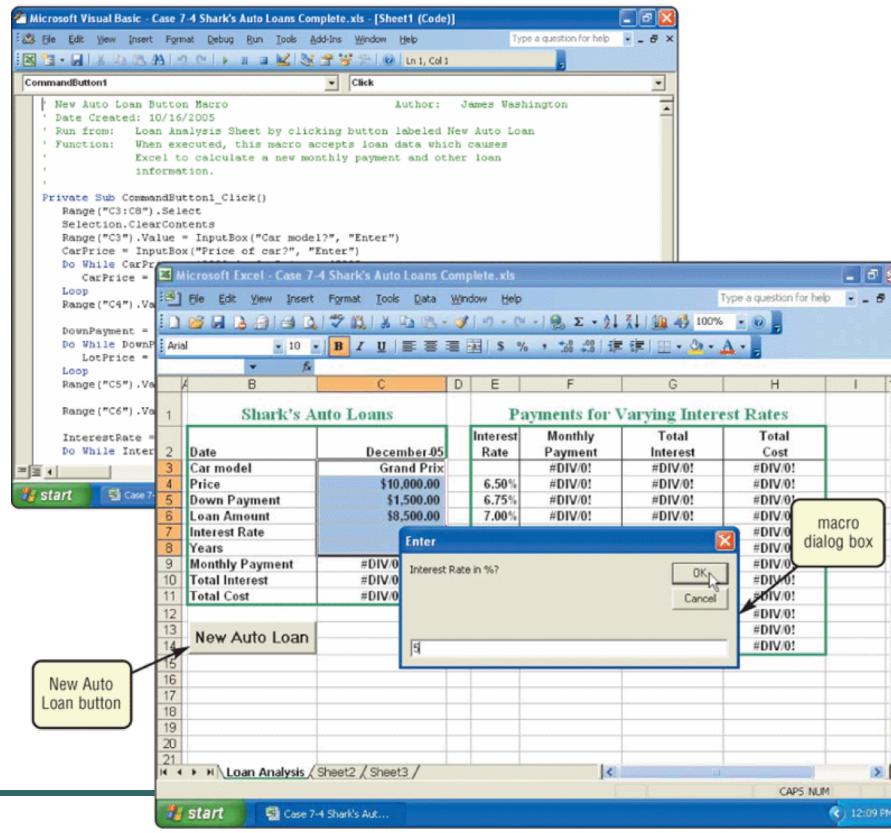
# Nonprocedural Languages and Program Development Tools

- **Application generator?**
- Là chương trình tạo source code hoặc machine code từ các specification
- Bao gồm các chương trình tạo Report , form, và tạo menu
  - Form cung cấp các vùng để vào dữ liệu

The image displays two side-by-side windows from a software application. The left window, titled 'Contacts : Form', shows a grid-based form design interface. It contains various fields such as 'First Name', 'Last Name', 'Company', 'Address', 'City', 'State/Province', 'Postal Code', 'Country/Region', 'Contact Type', 'Email Name', 'Referred By', 'Birthdate', and 'Notes'. Some fields have validation rules like '[FirstName] & " "' & [L]' entered into them. The right window, also titled 'Contacts', shows a preview of a contact record. The contact details are: First Name: Willie, Last Name: Popa, Company: Airlite Manufacturing, Dear: Willie, Address: P.O. Box 1192, City: Hammond, State/Province: IN, Postal Code: 46323, County/Region: USA, Contact ID: 1, Title: PA, Work Phone: (219) 555-0091, Work Extension: 3392, Mobile Phone: (219) 555-7261, Fax Number: (219) 555-2982. Below this, there is a summary section with fields: Contact Name (Willie Popa), Contact Type (dropdown menu), Email Name (popa@airlite.com), Referred By (JK), Birthdate (empty), and Notes (Customer for more than 20 years). Navigation buttons at the bottom include 'Calls...', 'Dial...', 'Page: 1 2', and 'Record: 1 2 \*'.

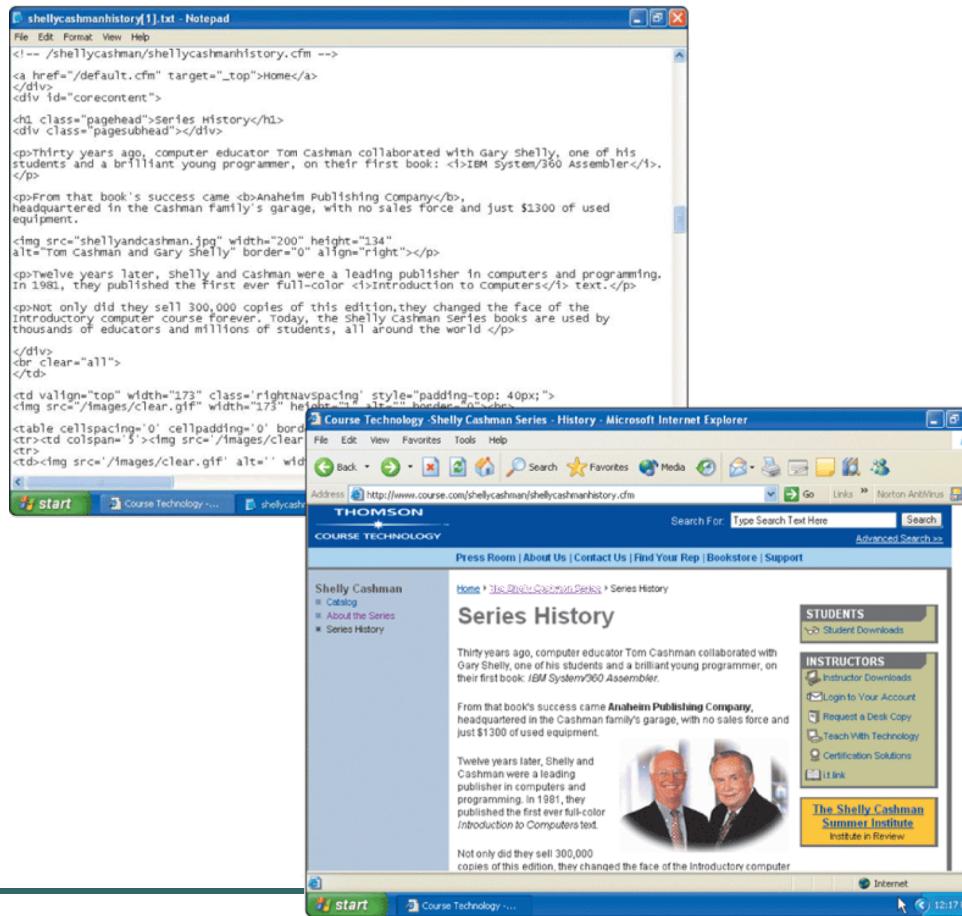
# Nonprocedural Languages and Program Development Tools

- Visual Basic for Applications (VBA)?
  - Macro programming language
    - Macro—Dãy các lệnh dùng để tự động hóa các công việc



# Web Page Development

- **HTML (Hypertext Markup Language)?**  
Dùng để tạo các trang Web



Next ▶

# Web Page Development

- Các hiệu ứng đặc biệt và các phần tử tương tác được thêm vào trang Web như nào?

## Script

Thông dịch chương trình chạy trên client

## Applet

thường chạy trên client, nhưng được biên dịch

## Servlet

applet chạy trên server

## ActiveX control

Là chương trình nhỏ chạy trên client

## Counter

đếm số người thăm Web site

## Image map

Hình ảnh đồ họa trả tới URL

## Processing form

Thu thập số liệu từ visitors

# Web Page Development

## ● Common gateway interface (CGI)?

Chuẩn giao tiếp xác định cách thức Web server giao tiếp với các nguồn tài nguyên bên ngoài

- CGI script—program quản trị việc gửi và nhận dữ liệu qua CGI

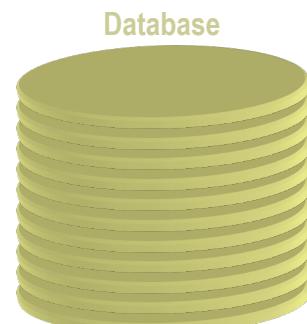
**Step 1.** LTV lưu các CGI program trong 1 thư mục đặc biệt trên Web server ví dụ /cgi-bin.



**Step 2.** Webmaster tạo 1 liên kết giữa CGI program và Web page. Khi 1 user hiện trang Web , CGI program sẽ automatically starts.



**Step 4.** CGI program nhận thông tin từ database, kết hợp chúng dưới dạng HTML , và gửi cho trình duyệt Web của User.



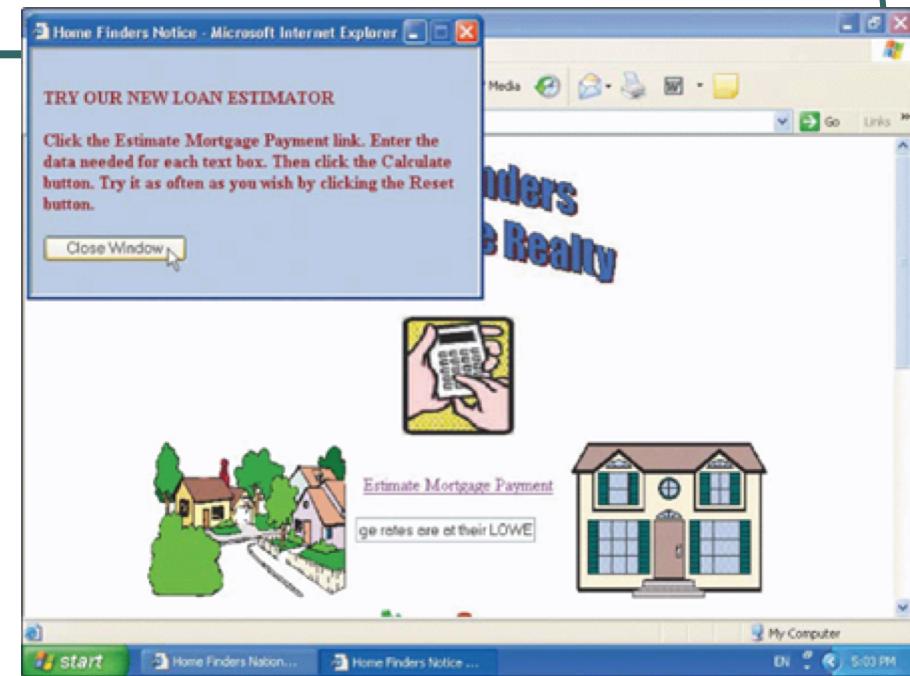
**Step 3.** Khi user khẳng định submits 1 yêu cầu, nó sẽ được gửi cho CGI program. CGI program kết nối với database và lấy các tin cho user. Ví dụ user yêu cầu xem phim *The Wizard of Oz*.



# Web Page Development

## ● Scripting language? DỄ HỌC VÀ SỬ DỤNG

- **JavaScript**—thêm các nội dung động và các phần tử tương tác vào Web page
- **VBScript** (Visual Basic, Scripting Edition)—Thêm tính thông minh và tương tác vào Web page
- **Perl** (Practical Extraction and Report Language)—Có khả năng xử lý văn bản rất mạnh



```
<HTML>
<HEAD>
<TITLE>Home Finders Notice</TITLE>
</HEAD>

<BODY bgColor="#B0C4DE">

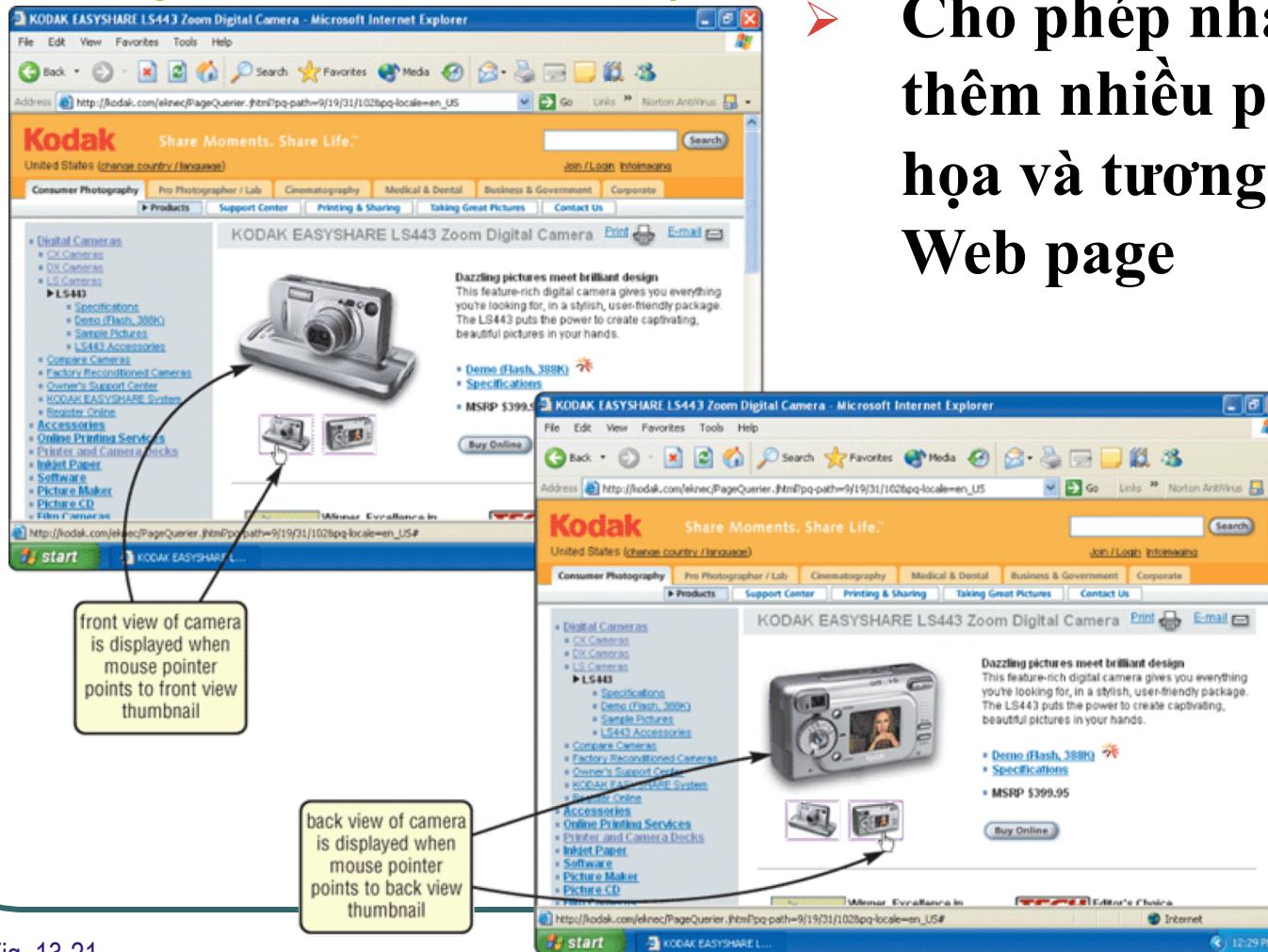
<P><BR>
<FONT COLOR="8b0000"><B>TRY OUR NEW LOAN ESTIMATOR</B></FONT>
<P>Click the Estimate Mortgage Payment link. Enter the
data needed for each text box. Then click the calculate
button. Try it as often as you wish by clicking the
Reset button.</B> </P>

<FORM>
<P><INPUT Type="Button" value="Close window" onclick="window.close()"> </P>
</FORM>
</FONT>
</BODY>
</HTML>
```

Next ▶

# Web Page Development

## ● Dynamic HTML (DHTML)?



p. 682 Fig. 13-21



Cho phép nhà phát triển  
thêm nhiều phần tử đồ  
họa và tương tác vào  
Web page

Next

# Web Page Development

## ● XHTML, XML, và WML?

### XHTML

(Extensible HTML)

tạo khả năng Web sites có thể hiện dễ ràng hơn trên các trình duyệt

Chứa các tính năng của HTML và XML

### XML

(Extensible Markup Language)

Cho phép developers có thể tạo các thẻ - tags – riêng của mình

Server gửi toàn bộ bản ghi cho client, tạo khả năng cho client có thể thực hiện việc xử lý mà không phải quay lại server

### WML

(Wireless Markup Language)

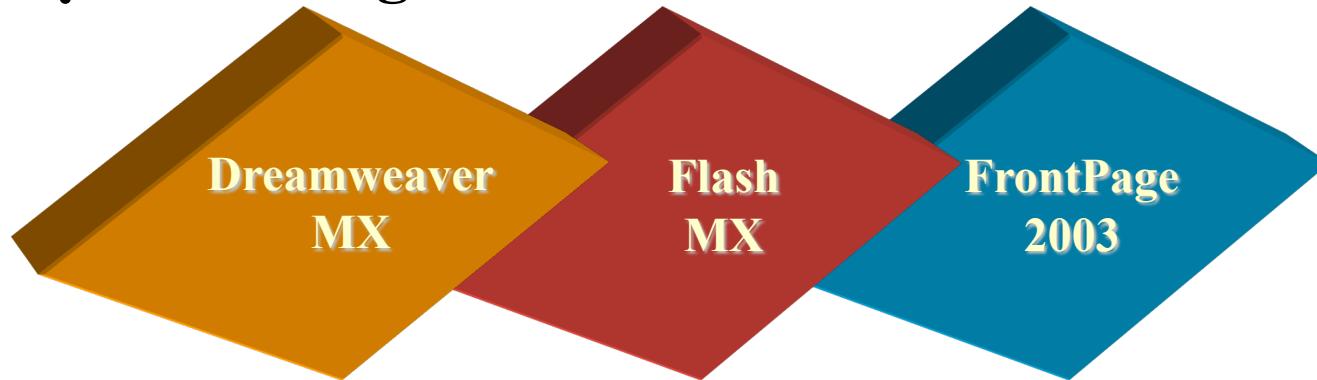
Cho phép developers có thể thiết kế những trang cho các trình duyệt chuyên dụng – mobil, ...

Sử dụng chuẩn wireless application protocol (WAP), để xác định cách thức các thiết bị không dây liên lạc với Web

# Web Page Development

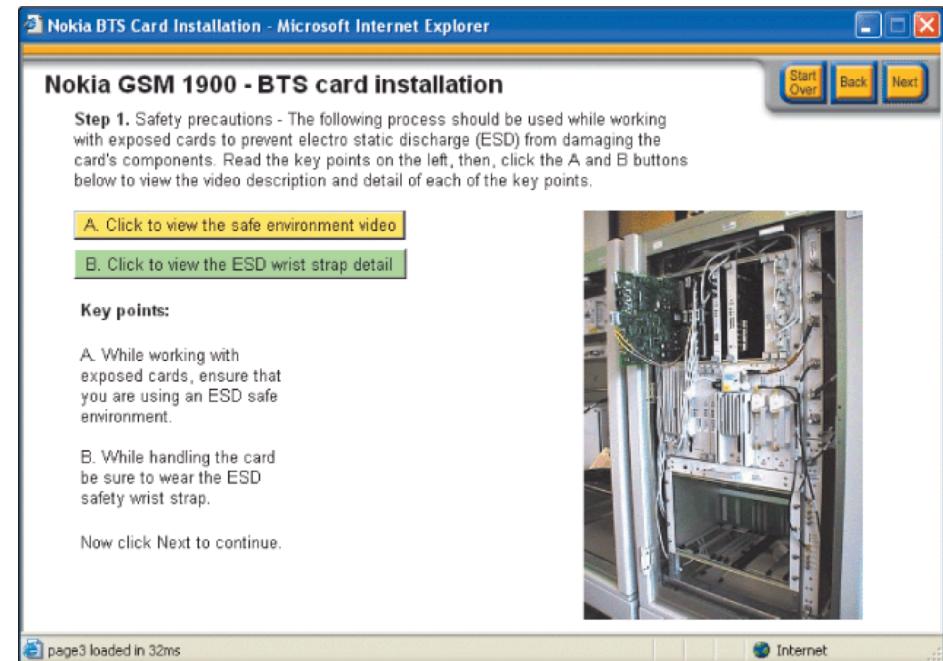
---

- **Web page authoring software?**
- **Tạo các trang Web hoàn hảo mà không cần dùng HTML**
- **Tự tạo các trang HTML**



# Multimedia Program Development

- **Multimedia authoring software?**
  - **Kết hợp văn bản, đồ họa, hoạt hình, âm thanh và video trong 1 bài trình diễn có tương tác**
  - **Sử dụng cho computer-based training (CBT) và Web-based training (WBT)**
  - **Software includes Toolbook, Authorware, và Director**



# Các Programming Languages khác

ADA

ALGOL

APL

FORTH

FORTRAN

HYPERTALK

LISP

LOGO

MODULA-2

PASCAL

PILOT

PL/I

PROLOG

SMALLTALK

# Python Courseware

---

## 1. Introduction to Computer Science and Programming in Python (2016)

- <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/index.htm>
- 12 Videos

# Python Courseware

---

## 2. Introduction to Computational Thinking and Data Science

- <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0002-introduction-to-computational-thinking-and-data-science-fall-2016/>
- 15 Videos