



# Bài 1: Tổng quan về lập trình Windows

Lương Trần Hy Hiến

FIT, HCMUP

*Lập trình Windows Form với C#*

# Giới thiệu môn học

## ■ Mục tiêu

- Cung cấp kiến thức và kỹ năng lập trình ứng dụng trong môi trường Windows cơ bản & nâng cao.
- Cung cấp kiến thức lập trình C# cho các ứng dụng Windows và các hàm thư viện cơ bản MS.NET.
- Giúp sinh viên làm quen với môi trường phát triển tích hợp hiện đại & mạnh mẽ của MS là Visual Studio .NET 2010, 2013, 2017.

# Giới thiệu môn học

- **Nội dung:** (ba nhóm kiến thức chính)
  - Trang bị kiến thức
    - Nền tảng về .NET Framework
    - Cơ bản về ngôn ngữ C#
  - Kiến thức về lập trình ứng dụng trong môi trường Windows với C#.
    - Xây dựng ứng dụng dựa trên Form (Windows Form Application)
  - Cung cấp các kiến thức nâng cao về lập trình cơ sở dữ liệu trong C#: ADO.NET/LINQ/EF

## Phục vụ cho môn học

- **Sinh viên cài đặt các tool sau**
  - MS .NET Framework 4.5+
  - MS Visual Studio .NET 2013 Pro (Express)
  - MS SQL Server 2012 (Express) (Option)

# Đề cương môn học

- **Phần 1: C# cơ bản – hướng đối tượng**
  - Môi trường .NET Framework
  - Ngôn ngữ C#: kiểu dữ liệu, biến, phạm vi hoạt động, toán tử, câu lệnh, điều khiển
    - Namespace, lớp thư viện
    - Lớp, phương thức, thừa kế...
    - Mảng, struct, enum
    - Giao diện, thuộc tính, indexer
    - Nhóm đối tượng (Array, Collection)
    - Delegate & event

## Đề cương môn học (2)

- **Phần 2: Lập trình ứng dụng trong môi trường Windows**
  - Xây dựng ứng dụng Windows Form (WinForm)
  - Windows Controls
  - Menu, context-sensitive menu, tool strip, status strip.
  - Xây dựng ứng dụng SDI, MDI
  - Dialog, common dialog
  - GDI+ (option)
  - User Control

## Đề cương môn học (3)

### ■ **Phần 3: Lập trình CSDL với ADO.NET**

- Tổng quan về ADO.NET
  - Cơ chế Disconnected...
- Kết nối CSDL MS SQL Server, MS Access
  - Chuỗi kết nối, đối tượng thực thi truy vấn dữ liệu
- Sử dụng DataSet để lưu trữ dữ liệu tách rời với data source
- Sử dụng chức năng Data Configuration Wizard để kết nối dữ liệu nhanh chóng
- Cơ chế data binding giữa control với data source

## Đề cương môn học (4)

- **Phần 4: Các phương pháp lập trình**
  - LINQ to SQL (Entity Framework)
  - Lập trình theo mô hình 3 lớp
  - Mô hình MVP



## Tài liệu [gợi ý] tham khảo

1. **Pro C# 7.0 with .NET and .NET Core, 8<sup>th</sup> edition.**
2. **Step by step MS ADO.NET Entity Framework, MS**
3. <http://www.codeproject.com>

# Đồ án môn học

- Đề tài: Phân tích, thiết kế và hiện thực chương trình quản lý theo hướng đối tượng
- Nhóm: 3-5 sinh viên
- Thời hạn: cuối kỳ (có nộp phác thảo GK)
- Hình thức: báo cáo + CD chương trình
- Gợi ý:
  - Quản lý bán hàng
  - Quản lý sinh viên
  - Quản lý thư viện
  - Quản lý nhân sự
  - .....

# Nội dung báo cáo

- I. Giới thiệu chương trình
- II. Cơ sở dữ liệu
- III. Các chức năng (Demo)

## 5 lời khuyên để đạt kết quả tốt



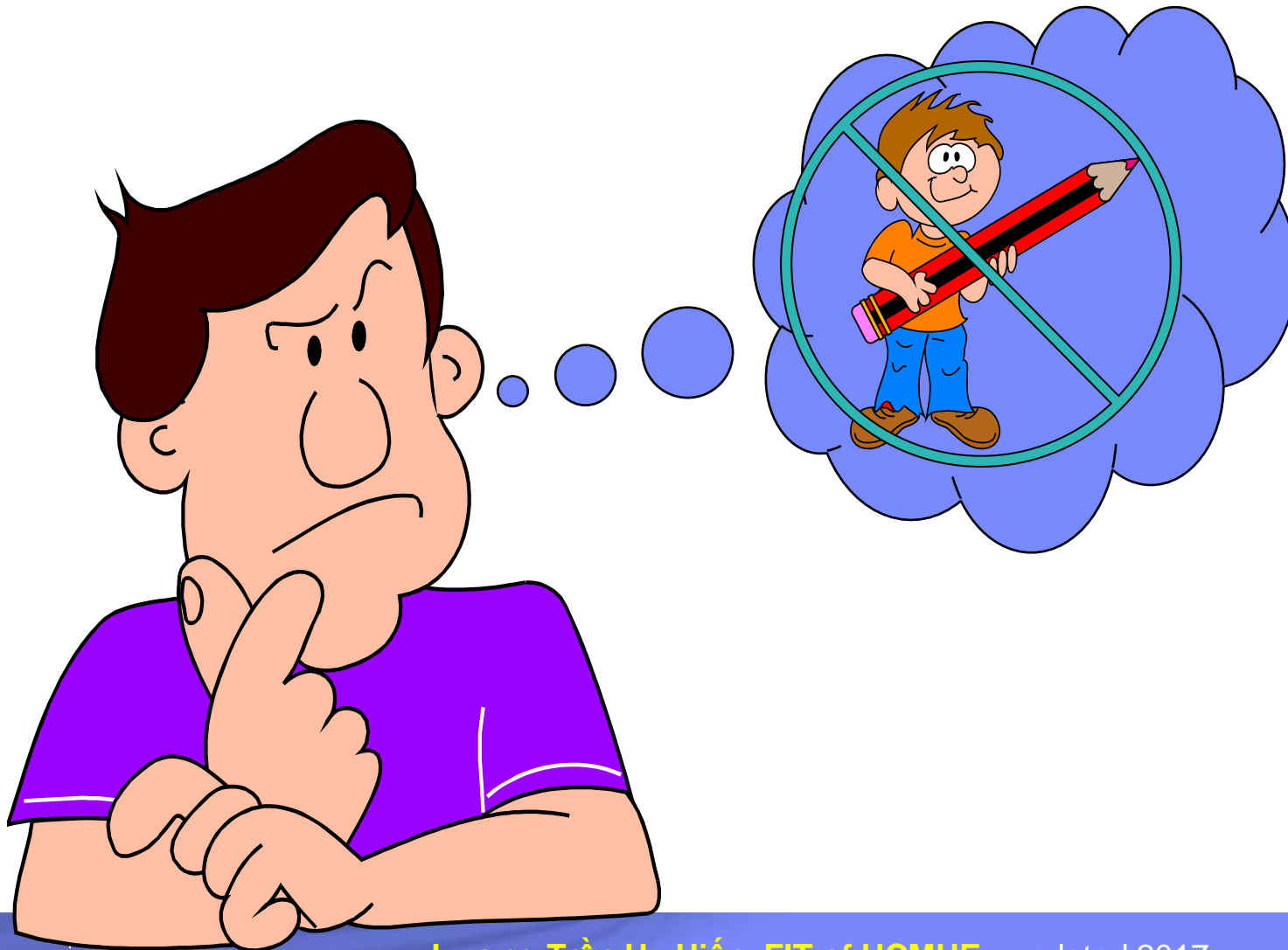
# Làm việc chăm chỉ



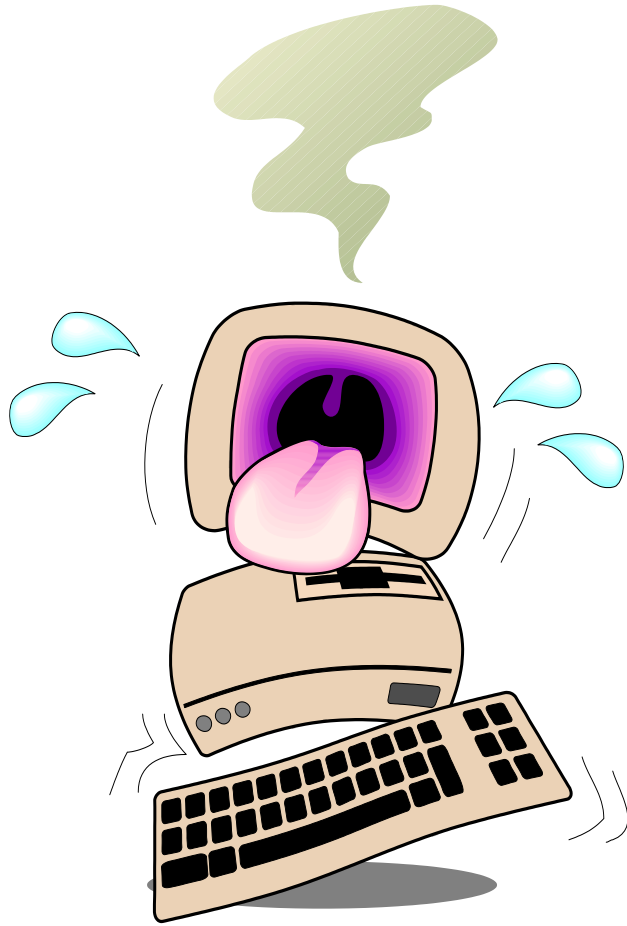
# Làm bài tập và thực hành nhiều



# Tự làm các bài tập và thực hành

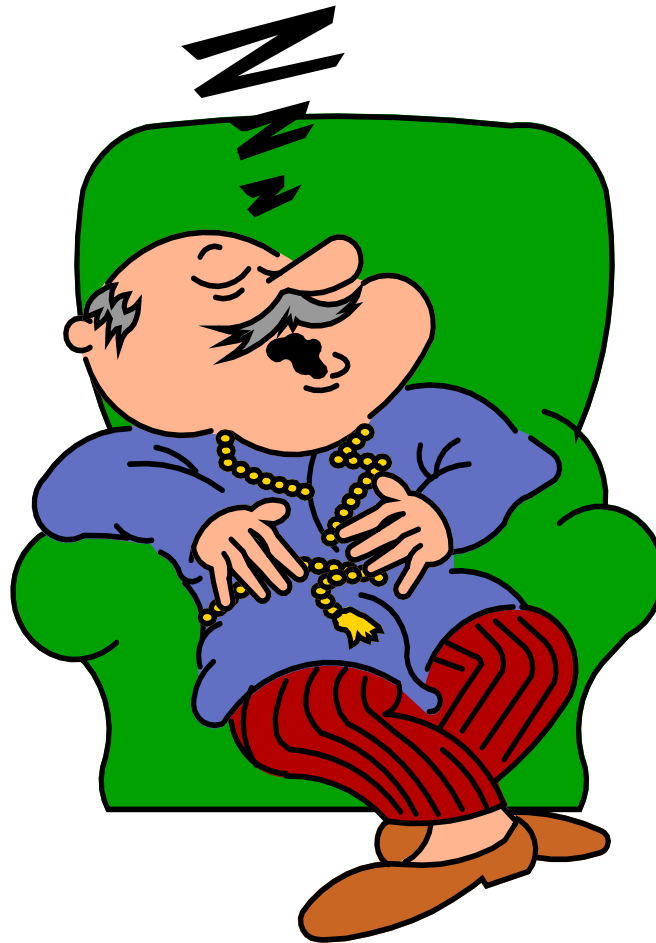


# Kiên nhẫn với máy tính

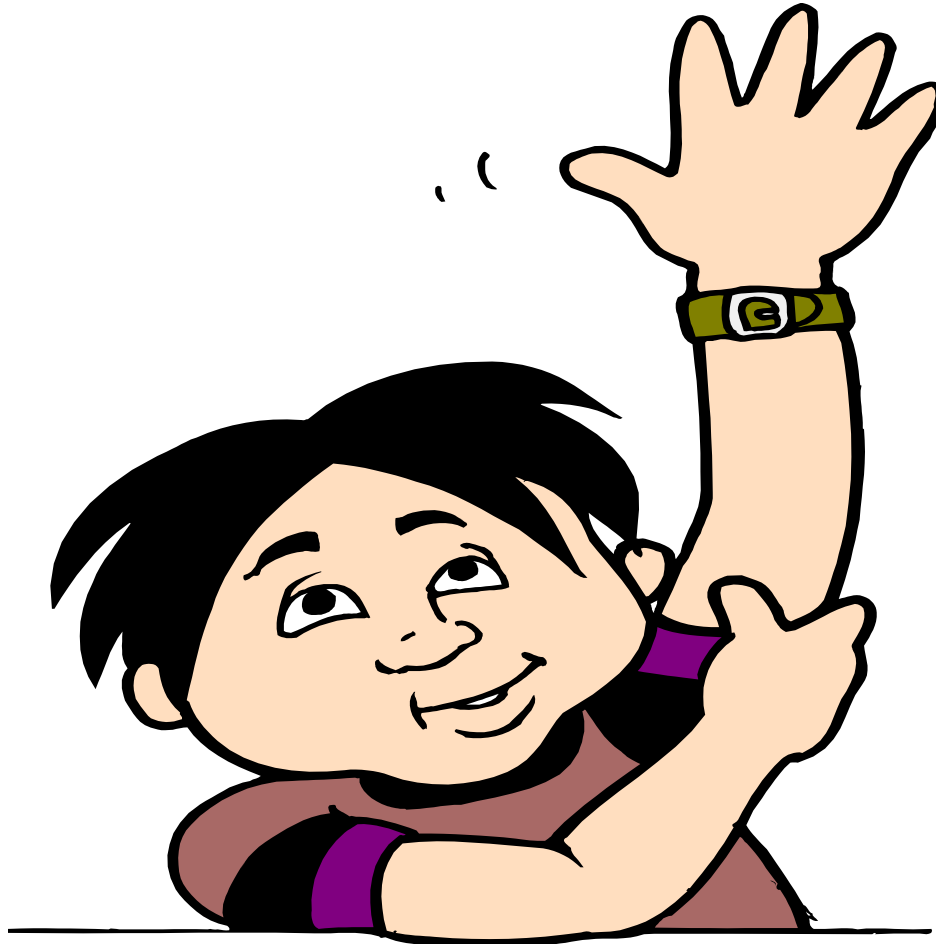




Nếu cần thiết, **ngủ một tí ...**



# Hỏi và Đáp



# Môi trường Windows

- Hệ điều hành **32/64 bits**
- Giao diện người dùng kiểu đồ họa (**GUI**)
- **Đa nhiệm**

# Giao diện người dùng kiểu đồ họa (GUI)

- **Visual Interface**
- **WYSIWYG**
- **Các ứng dụng Windows có giao diện thống nhất với cùng dáng vẽ:**
  - một cửa sổ hình chữ nhật
  - thanh tiêu đề
  - menu, hộp hội thoại, thanh trượt

## Đa nhiệm

- Nhiều chương trình có thể được kích hoạt và chạy cùng một lúc
- Mỗi chương trình chiếm một phần tài nguyên của hệ thống và có một phần bộ nhớ thường trú
- Windows 32 bits hoạt động theo cơ chế preemptive. Mỗi chương trình có thể được tách thành các thread thực hiện cùng lúc

## Quản lý bộ nhớ

- Mã chương trình và dữ liệu có thể dịch chuyển trong bộ nhớ vật lý
- Khả năng sử dụng bộ nhớ phụ
- Các thư viện liên kết động (.dll)

## Tư tưởng hướng đối tượng

- Cửa sổ ứng dụng chính và các cửa sổ con là các thể hiện (đối tượng) của lớp cửa sổ được đăng ký trước.

# Giao diện đồ họa độc lập thiết bị

- **Giao diện đồ họa của Window cho phép tạo các đối tượng đồ họa và định dạng văn bản cho cả màn hình và máy in.**
- **Chương trình viết cho Windows không điều khiển trực tiếp thiết bị xuất như màn hình hay máy in mà thông qua một ngôn ngữ lập trình đồ họa gọi là Giao Diện Thiết Bị Đồ Hoạ (Graphics Device Interface: GDI).**



# Kiến trúc hướng thông điệp (message driven)

- Windows và các ứng dụng Windows hoạt động theo cơ chế truyền, nhận thông điệp
- Các hoạt động của chương trình thay đổi tùy theo thông điệp mà nó nhận được
- Thông điệp được gửi qua lại giữa ứng dụng và Windows, giữa các ứng dụng với nhau
- Cửa sổ ứng dụng tự động được vẽ lại mỗi khi có sự thay đổi kích thước hay vùng bị che. Điều này được thực hiện nhờ hệ điều hành gửi thông điệp cho chương trình

# Windows vs DOS

<b>Windows</b>	<b>DOS</b>
Lập trình sự kiện, dựa vào sự kiện (message)	Thực hiện tuần tự theo chỉ định
Multi-talking	Single task
Multi-CPU	Single CPU
Tích hợp sẵn Multimedia	Phải dùng các thư viện Multimedia riêng
Hỗ trợ 32 bits (hơn nữa)	Ứng dụng 16 bits
DLL, OLE, DDE, COM, OpenGL, DirectX	Không có

# Viết 1 ứng dụng Windows dựa trên ngôn ngữ tựa C

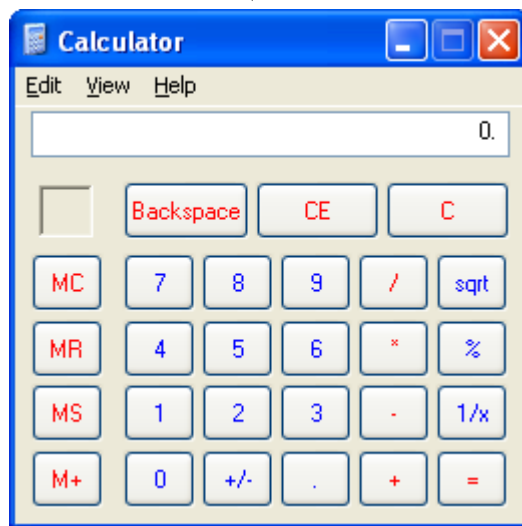
Năm	Ngôn ngữ	Công cụ
1985	C	Windows application programming interface (API)
1992	C++	Microsoft Foundation Class (MFC) Library
2001	C# hay C++.NET	Windows Form (1 phần của .NET Framework)

# Các thành phần giao diện đồ họa - GUI

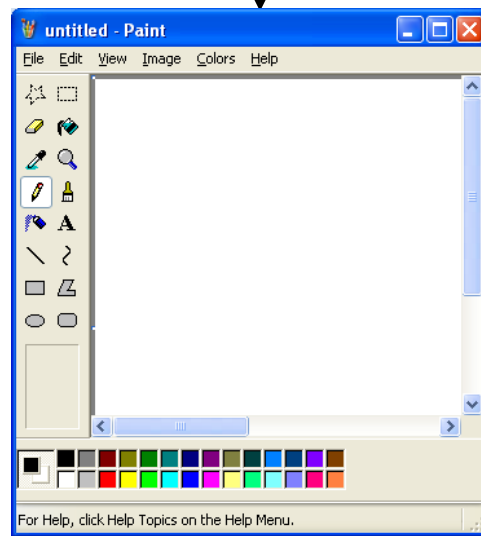
❖ GUI : Graphics User Interface

❖ Các dạng GUI cơ bản

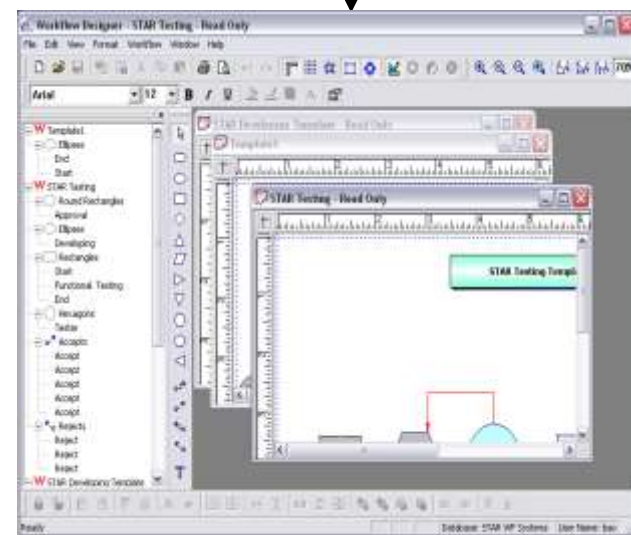
**Dialog**



**SDI**



**MDI**



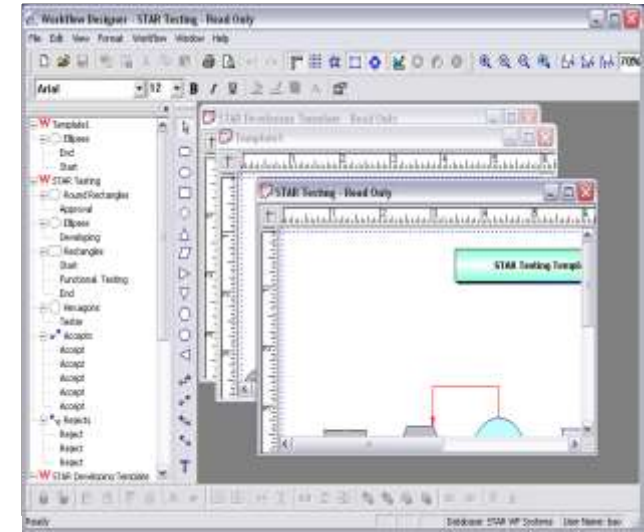
# Các thành phần giao diện đồ họa - GUI

## ❖ Dialog

- Có một cửa sổ làm việc
- Thường có kích thước cố định
- Thường không có menu bar

## ❖ SDI – Single Document Interface

- Có một cửa sổ làm việc
- Cho phép thay đổi kích thước cửa sổ (Resizable)
- Không có cửa sổ con



## ❖ MDI – Multiple Document Interface

- Có một cửa sổ làm việc chính (Main frame) và nhiều cửa sổ con (Child Frame)
- Cho phép thay đổi kích thước cửa sổ (Resizable)
- Cho phép Minimize/Maximize/Close/Arrange/Cascade cửa sổ con

Calculator

Paint

Workflow Editor

# GUI

- Định nghĩa:

- là 1 vùng chữ nhật trên màn hình,
- dùng để hiển thị kết quả output,
- và nhận các input từ người dùng

=> Công việc đầu tiên của 1 ứng dụng GUI là tạo 1 cửa sổ làm việc

- Nguyên tắc quản lý:

- Mô hình phân cấp: mỗi cửa sổ đều có 1 cửa sổ cha (parent window), ngoại trừ cửa sổ nền Desktop
- Tại mỗi thời điểm, chỉ có 1 cửa sổ nhận input từ user  
(Active/Focused window)

# Các thành phần giao diện đồ họa - GUI

**App Window**

**Desktop Window**

**Shortcut Icon**

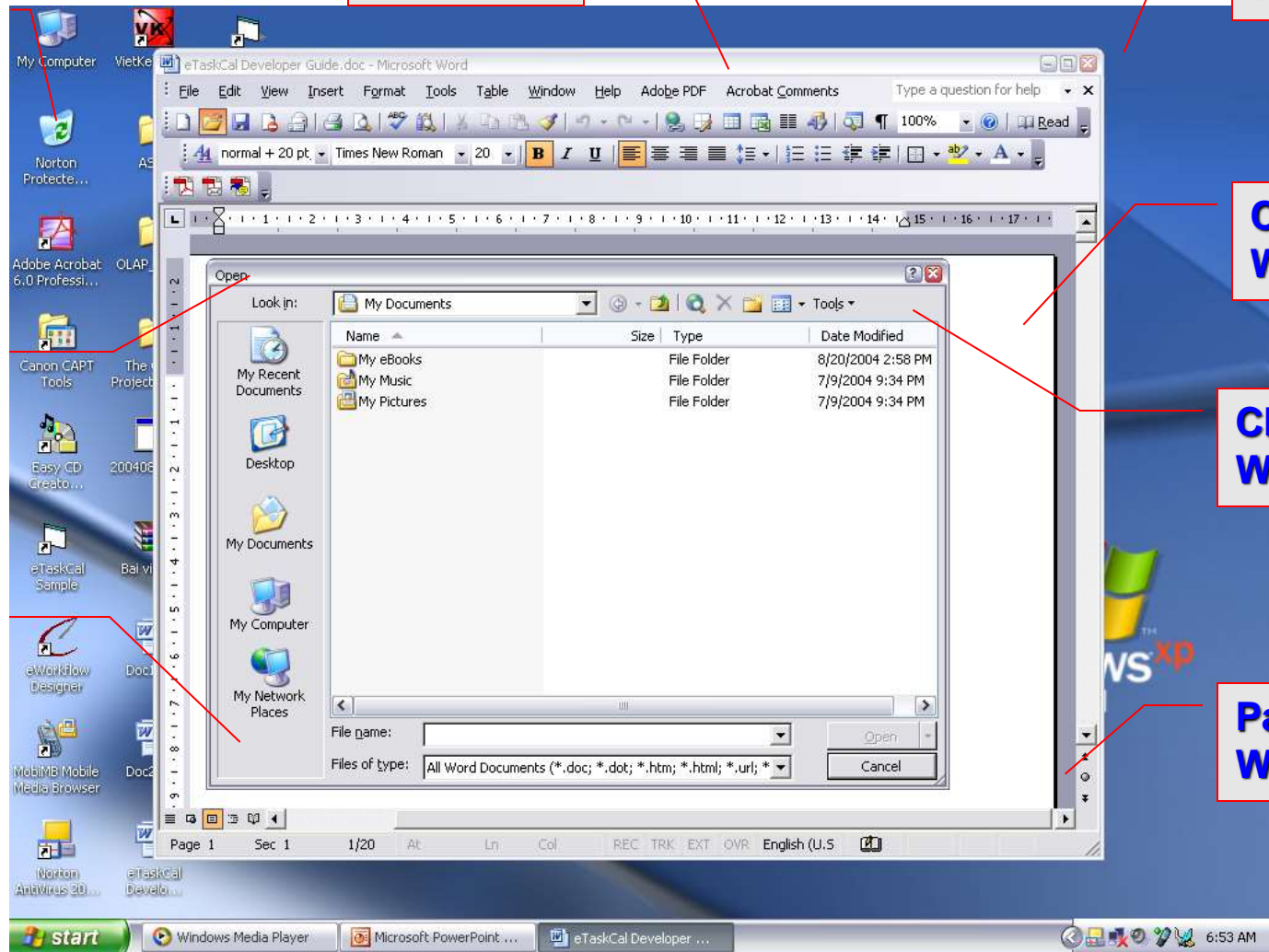
**Child Window**

**Dialog box**

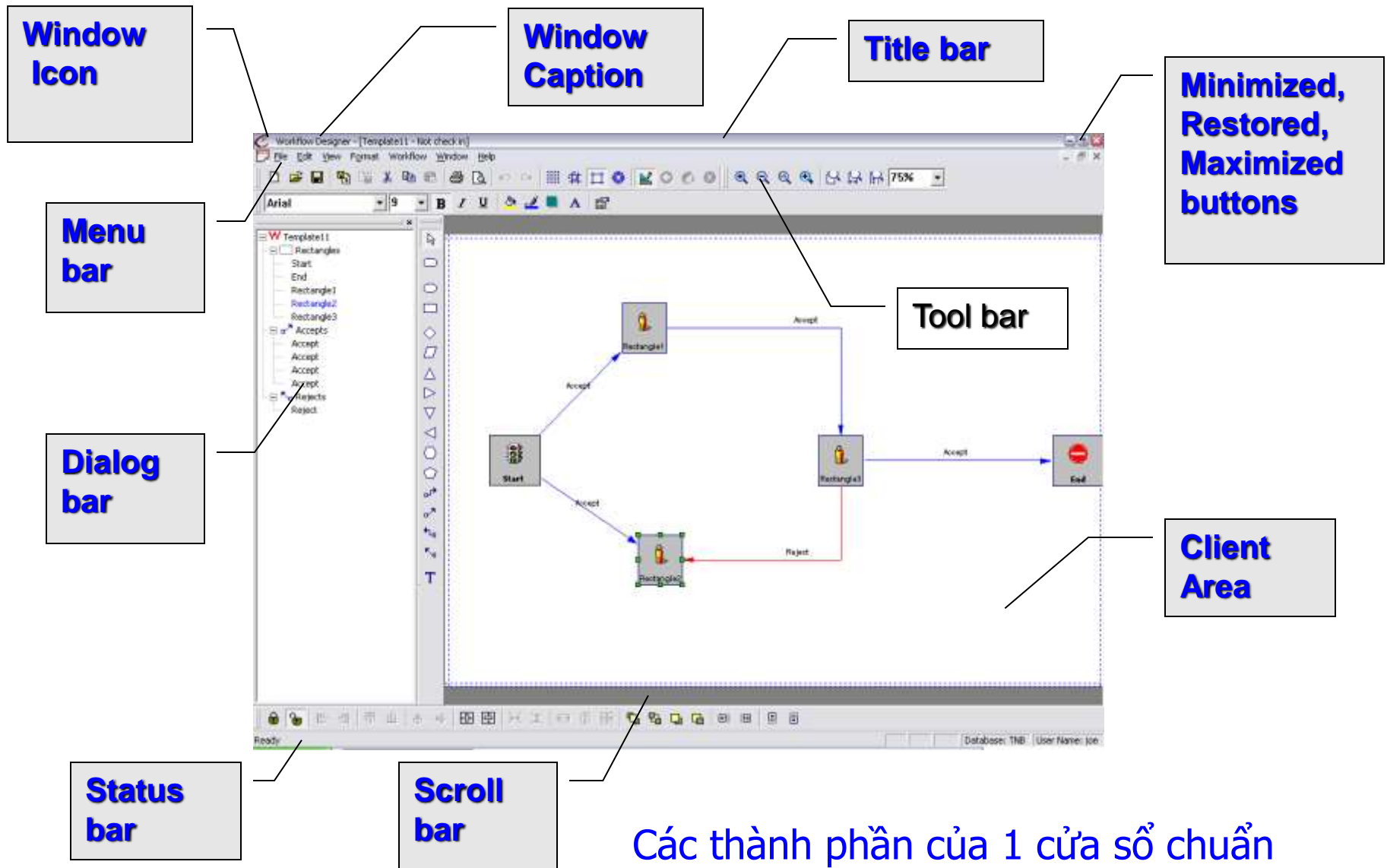
**Child Window**

**Control**

**Parent Window**



# Các thành phần giao diện đồ họa - GUI

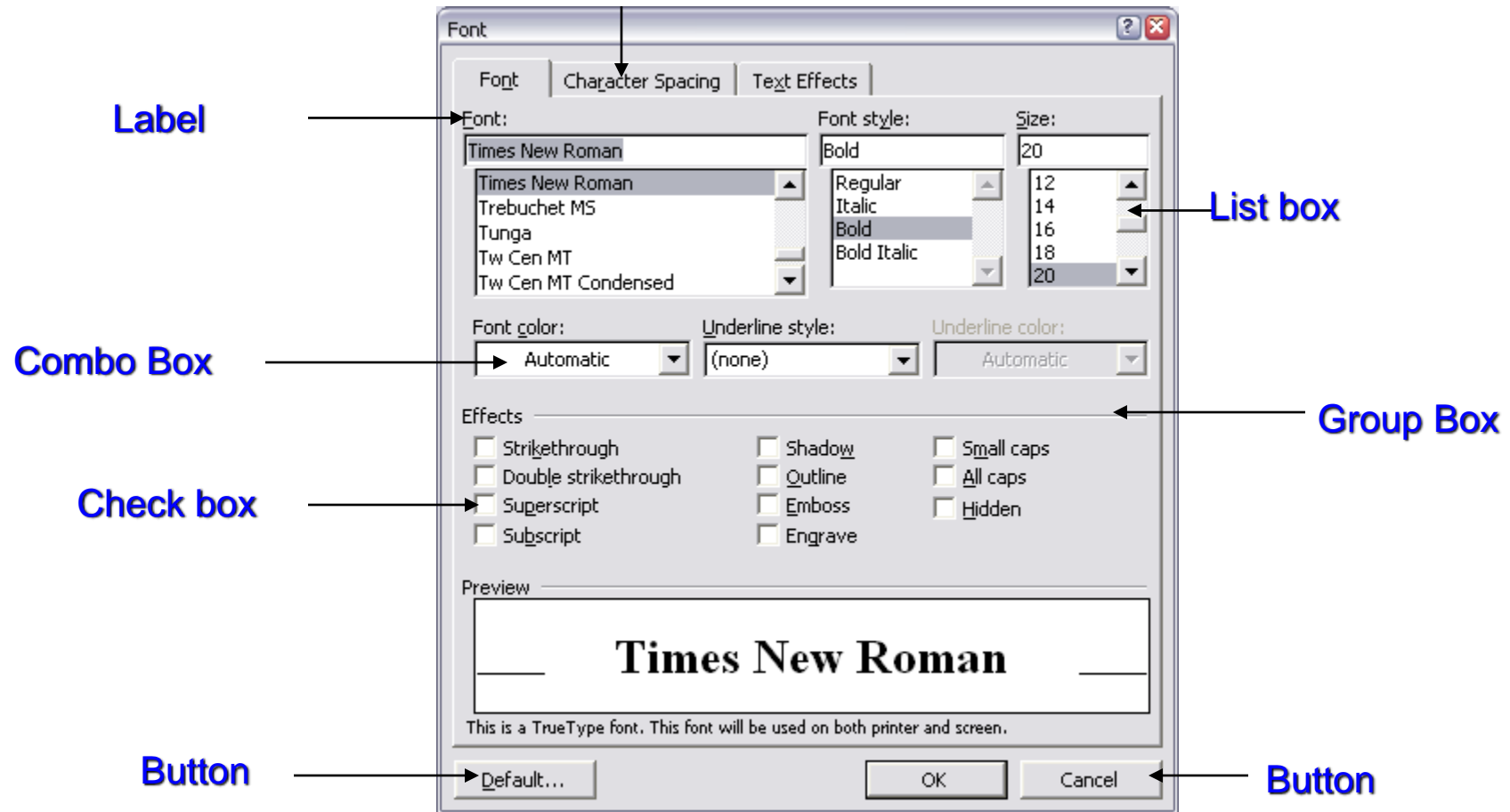


Các thành phần của 1 cửa sổ chuẩn



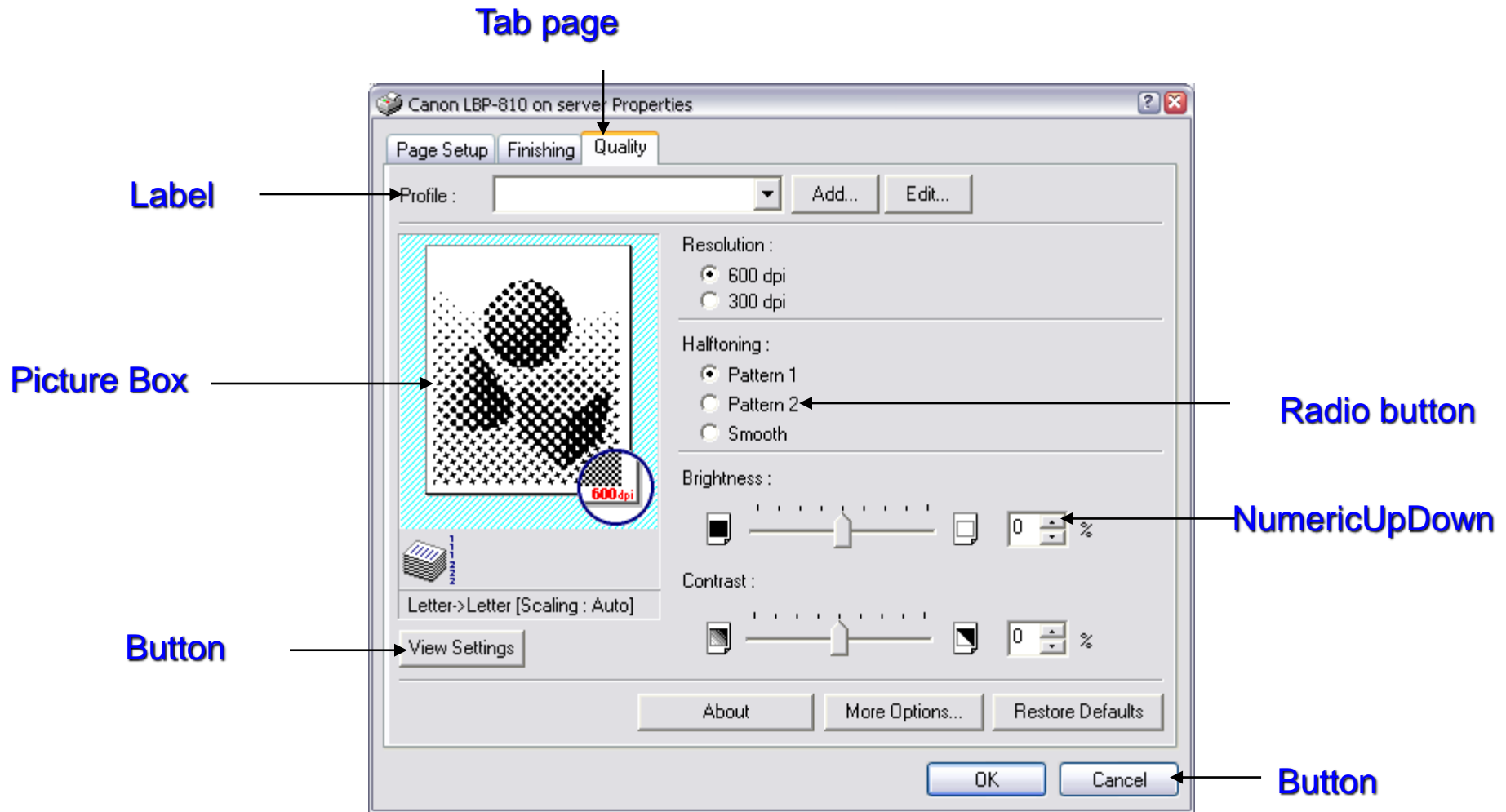
# GUI – Controls

Tab Page



## Các dạng control chuẩn của windows

# GUI – Controls



## Các dạng control chuẩn của windows

# GUI – Controls

