

# Window Programming

Visual C++ MFC Programming

Lecture 01

김예진

Dept. of Game Software

# Announcement

---

- 03/02: 501 → 502 등록

# Plan

---

- Window Programming 소개
- Visual C++ IDE (통합 개발 환경)
- 윈도우 응용 프로그램 개발 도구

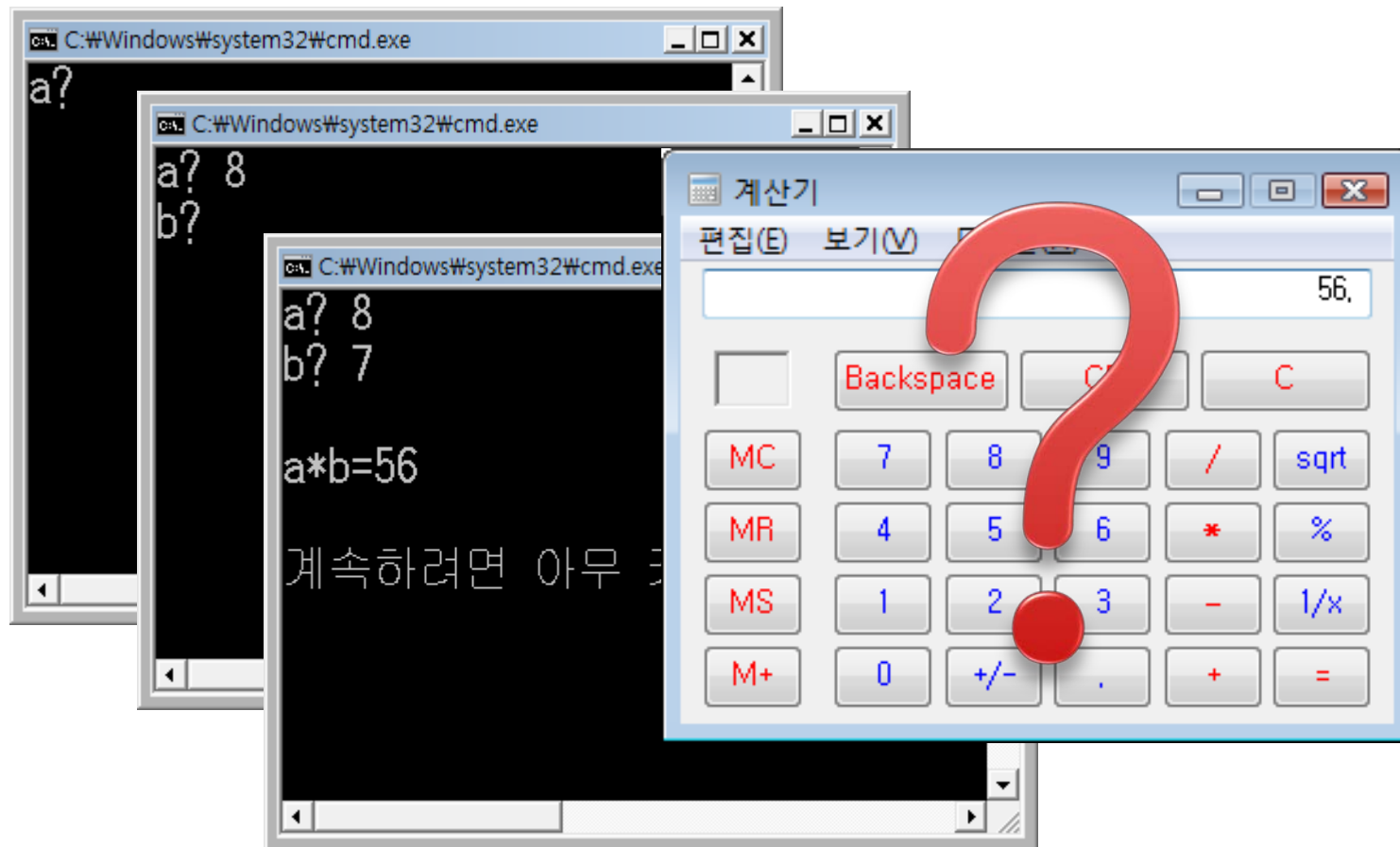
# Window Programming 소개

What is Visual C++?



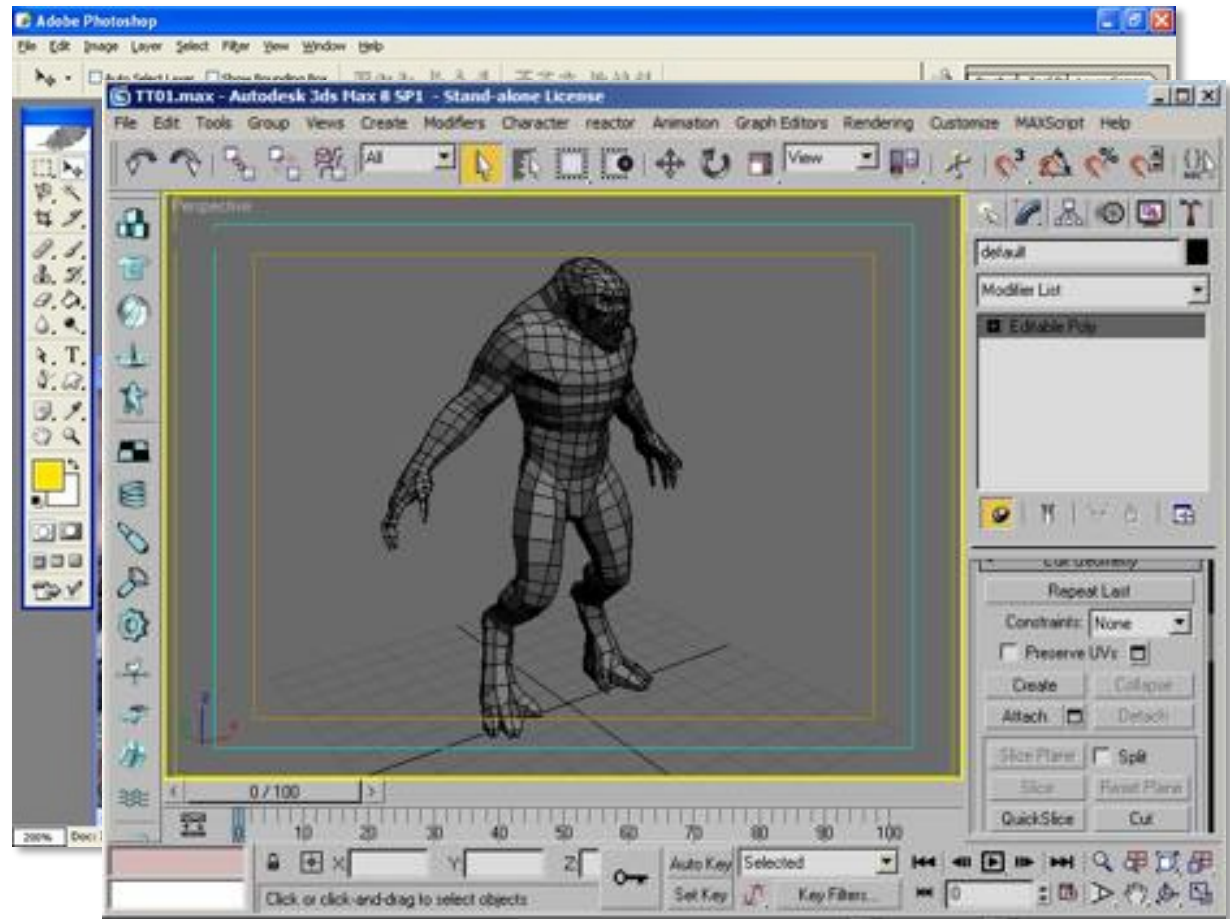
# 지금까지 해왔던 코딩

- C/C++ 프로그래밍



# 지금까지 해왔던 코딩

- C/C++ 프로그래밍
  - 변수/배열
  - =, +, -, \*, /
  - if
  - for
  - function
  - Pointer
  - **class**
  - **template**



# C++와 Windows 프로그래밍

---

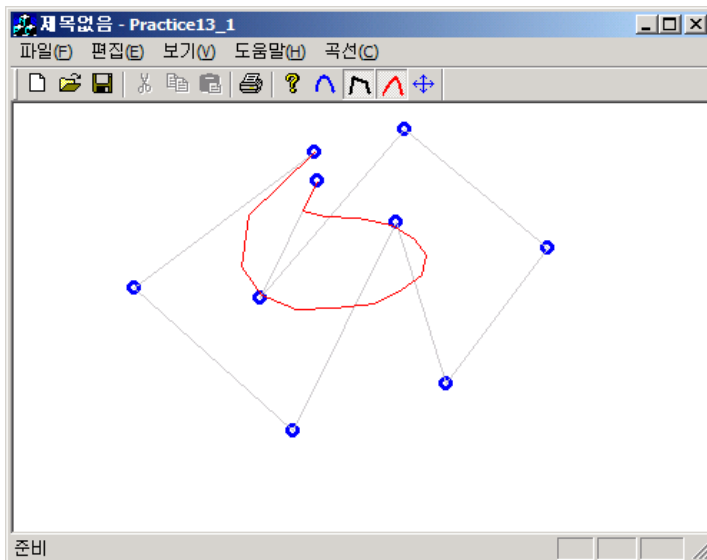
- C++의 충분한 이해 필요
  - 객체지향 언어 (Object Oriented Language)
  - 클래스 (Class), 상속 (Inheritance), 다형성 (Polymorphism), 가상 함수 (Virtual Functions)
- 자신만의 친숙한 개발 도구 선택
  - Borland C++ Builder → old
  - Microsoft Visual Studio (5.0~6.0, 2002~2013, 2015~2017)
- 본 과목에서는

Microsoft사의 **Visual Studio 2015** 에서 제공하는 IDE (Integrated Development Environment, 통합 개발 환경) 사용

# Windows 프로그램의 개념

- 윈도우 프로그래밍이란?
  - 사용자가 발생시키는 **이벤트**에 대한 **\*메시지**를 처리하는 것
  - 메시지 기반 (Message-driven) 프로그래밍
  - 이벤트 기반 (Event-driven) 프로그래밍

\*메시지: 프로그램의 외부 또는 내부의 변화를 해당 프로그램에 알리는 개념
- Windows 프로그램 vs. DOS 프로그램 ??
  - 코드의 실행 순서에 따른 차이



```
C:\cygwin\home\hoon\work\netsim>dir/w
Volume in drive C is IBM_PRELOAD
Volume Serial Number is 40BA-B69E

Directory of C:\cygwin\home\hoon\work\netsim

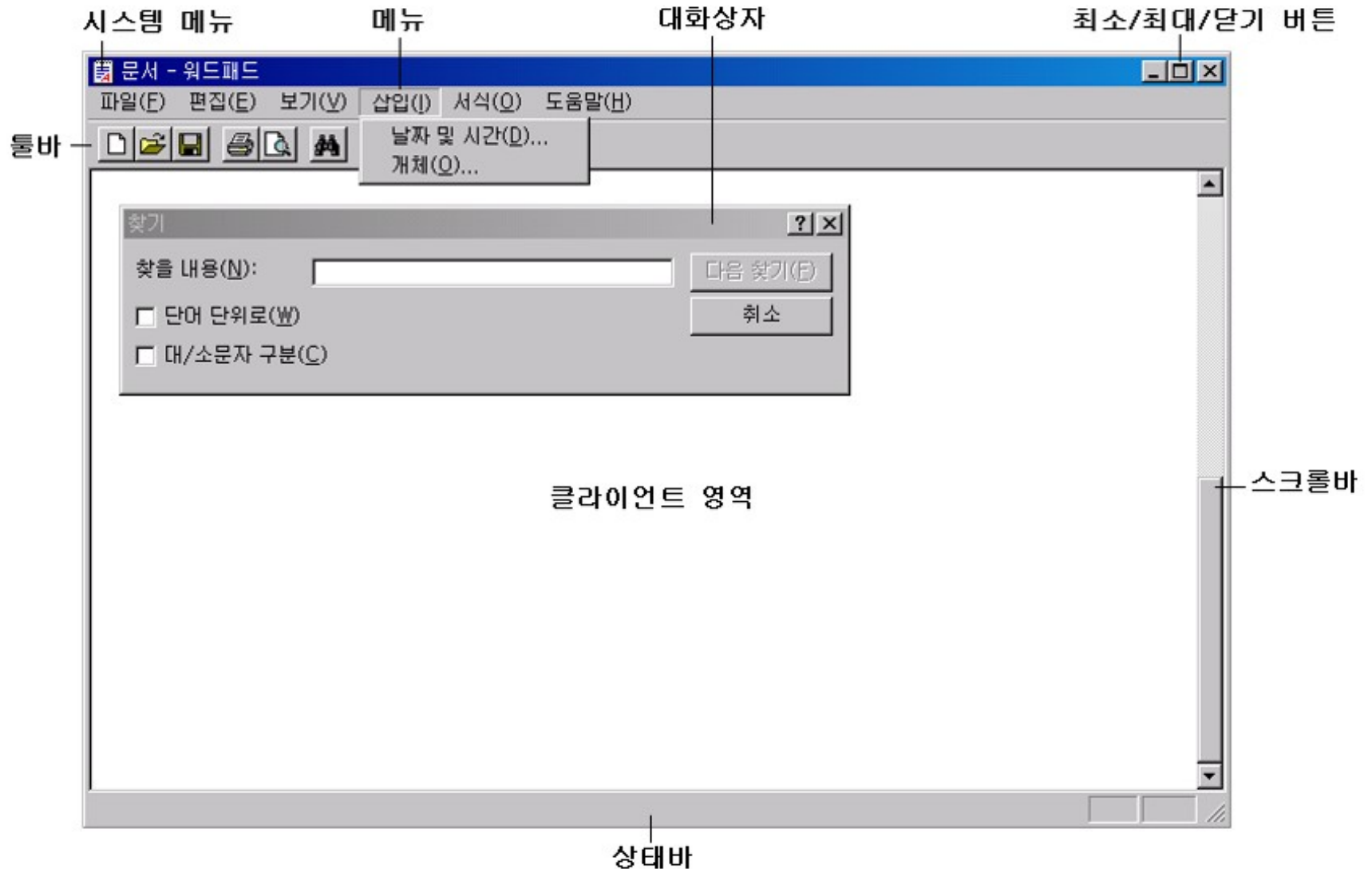
[.]           [..]           a.exe         [data]        [include]     [lib]
Makefile      Netsim.exe    netsim.IAB    netsim.IAD    netsim.IMB    netsim.IMD
netsim.PFI   netsim.PO    netsim.PR     netsim.PRI    netsim.PS     netsim.WK3
[src]

             13 File(s)          109,256 bytes
              6 Dir(s)  11,702,415,360 bytes free

C:\cygwin\home\hoon\work\netsim>netsim
Start netsim
Total coordinate X = 1500, Y = 300
Success in init the map.
node[0] => ( x = 433, y = 96, CID = 1 )
node[1] => ( x = 641, y = 197, CID = 7 )
node[2] => ( x = 209, y = 103, CID = 5 )
node[3] => ( x = 969, y = 53, CID = 3 )
node[4] => ( x = 821, y = 202, CID = 12 )
node[5] => ( x = 290, y = 81, CID = 0 )
node[6] => ( x = 393, y = 182, CID = 6 )
node[7] => ( x = 1069, y = 274, CID = 13 )
node[8] => ( x = 93, y = 180, CID = 5 )
node[9] => ( x = 1143, y = 17, CID = 3 )
node[10] => ( x = 1292, y = 80, CID = 4 )
node[11] => ( x = 603, y = 73, CID = 2 )
node[12] => ( x = 362, y = 236, CID = 11 )
```

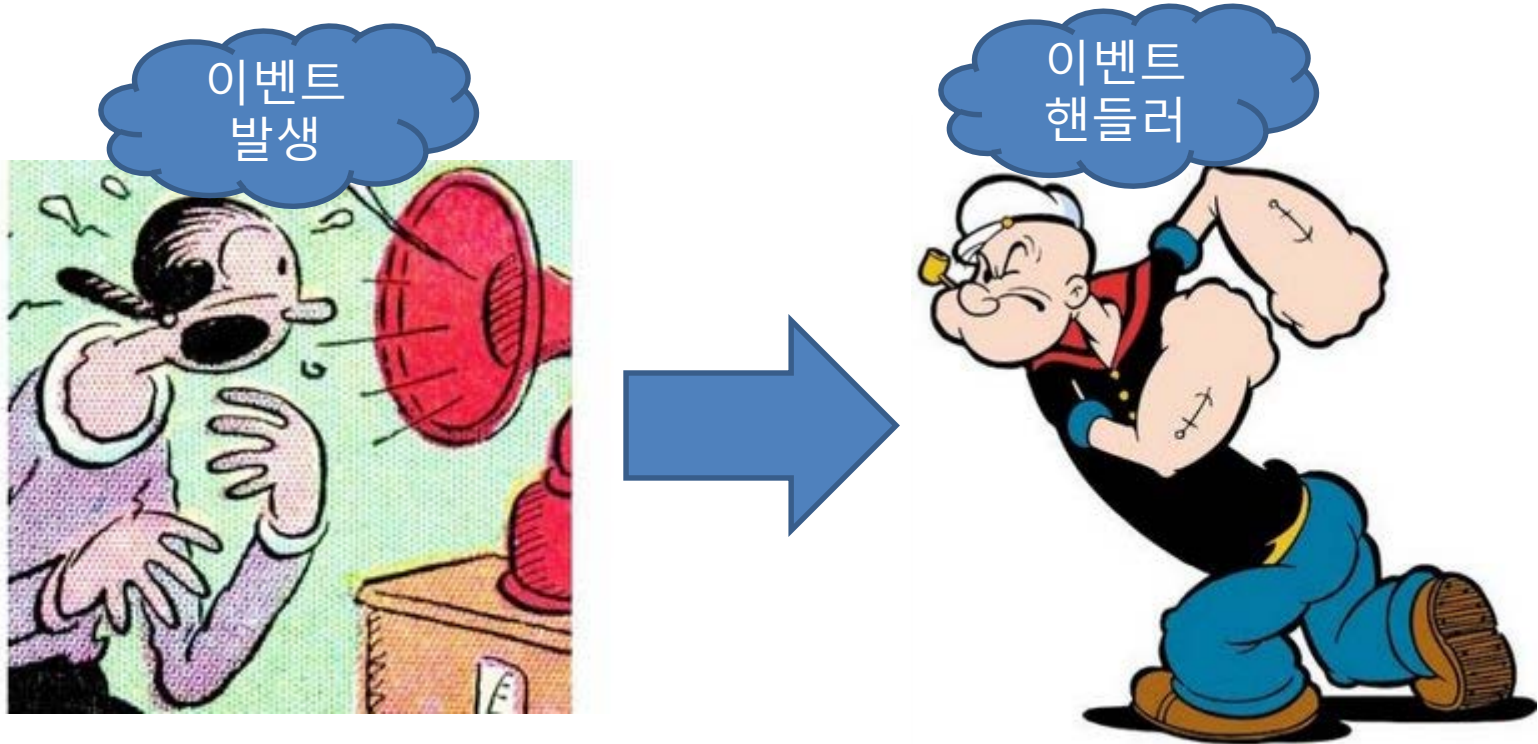


# GUI (Graphical User Interface)



# GUI Programming?

- 이벤트가 발생되었을 때 이를 처리하는 방법을 알려 주는 것



# 이벤트(Event)?

- 입력장치의 상태 변화 또는 내부의 상태변화

## MOUSE EVENT HANDLERS

starring **HANDLERMAN**



drag out



drag over



roll out



release outside



press

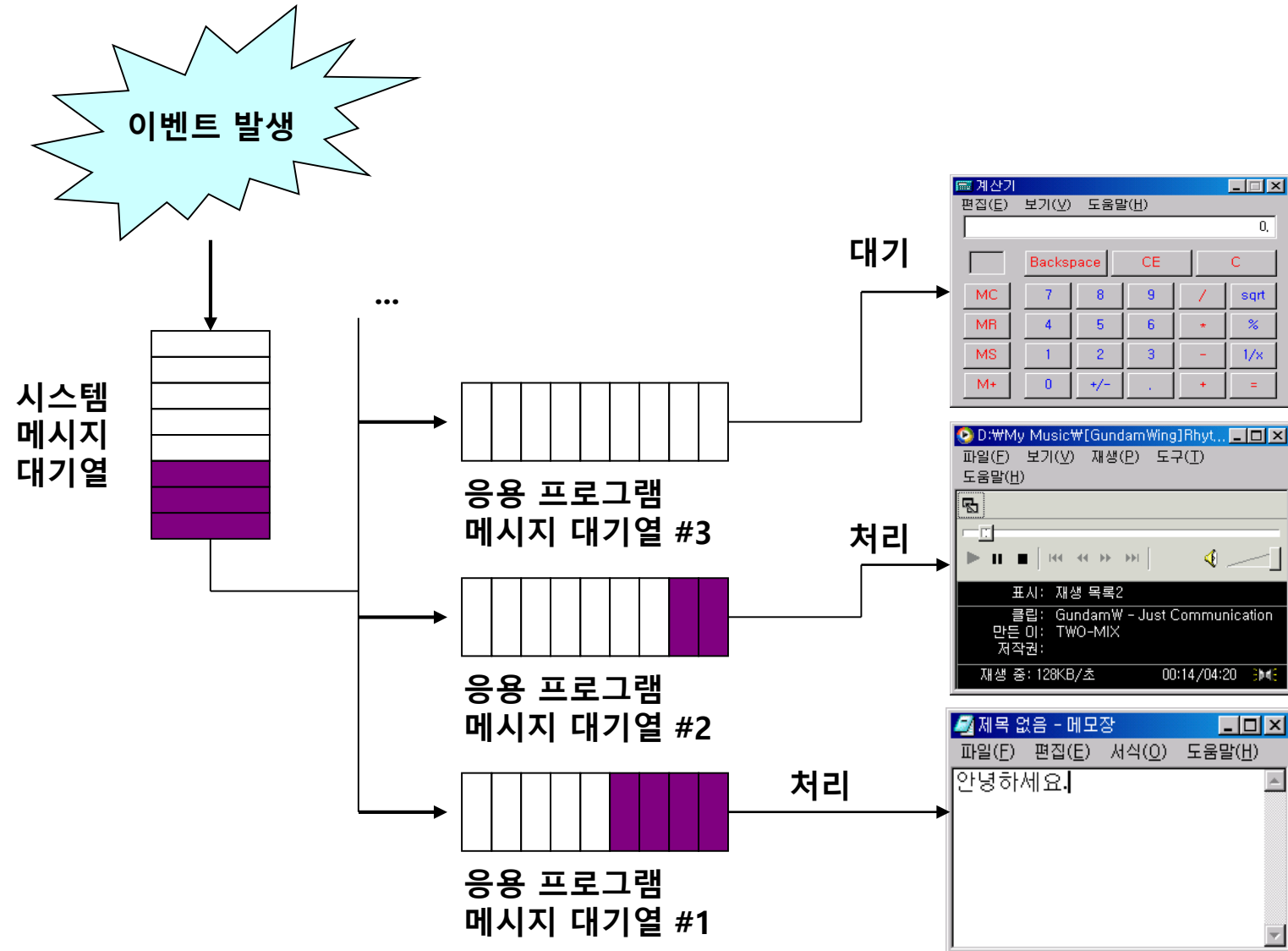


release

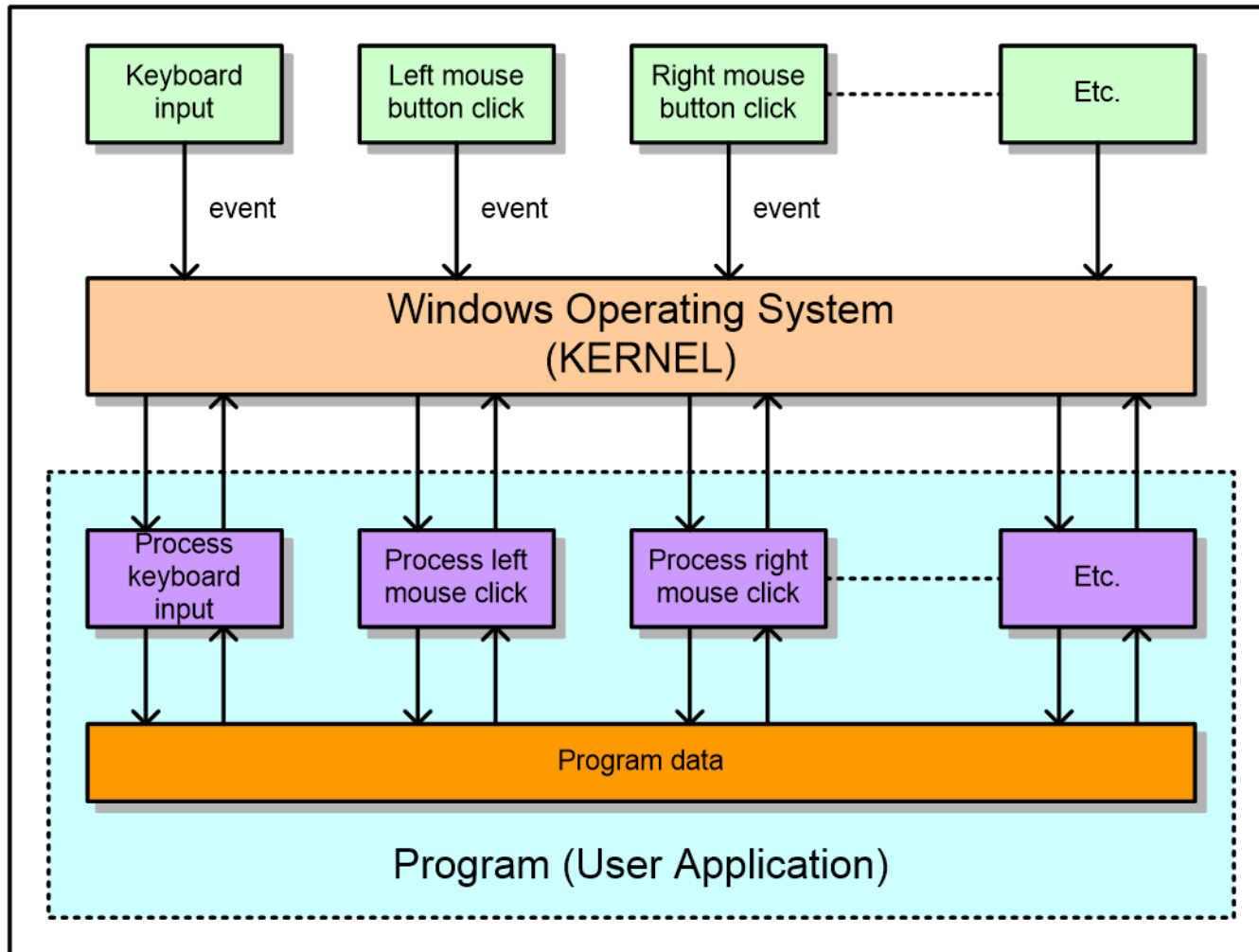


roll over

# 이벤트와 이벤트 핸들러

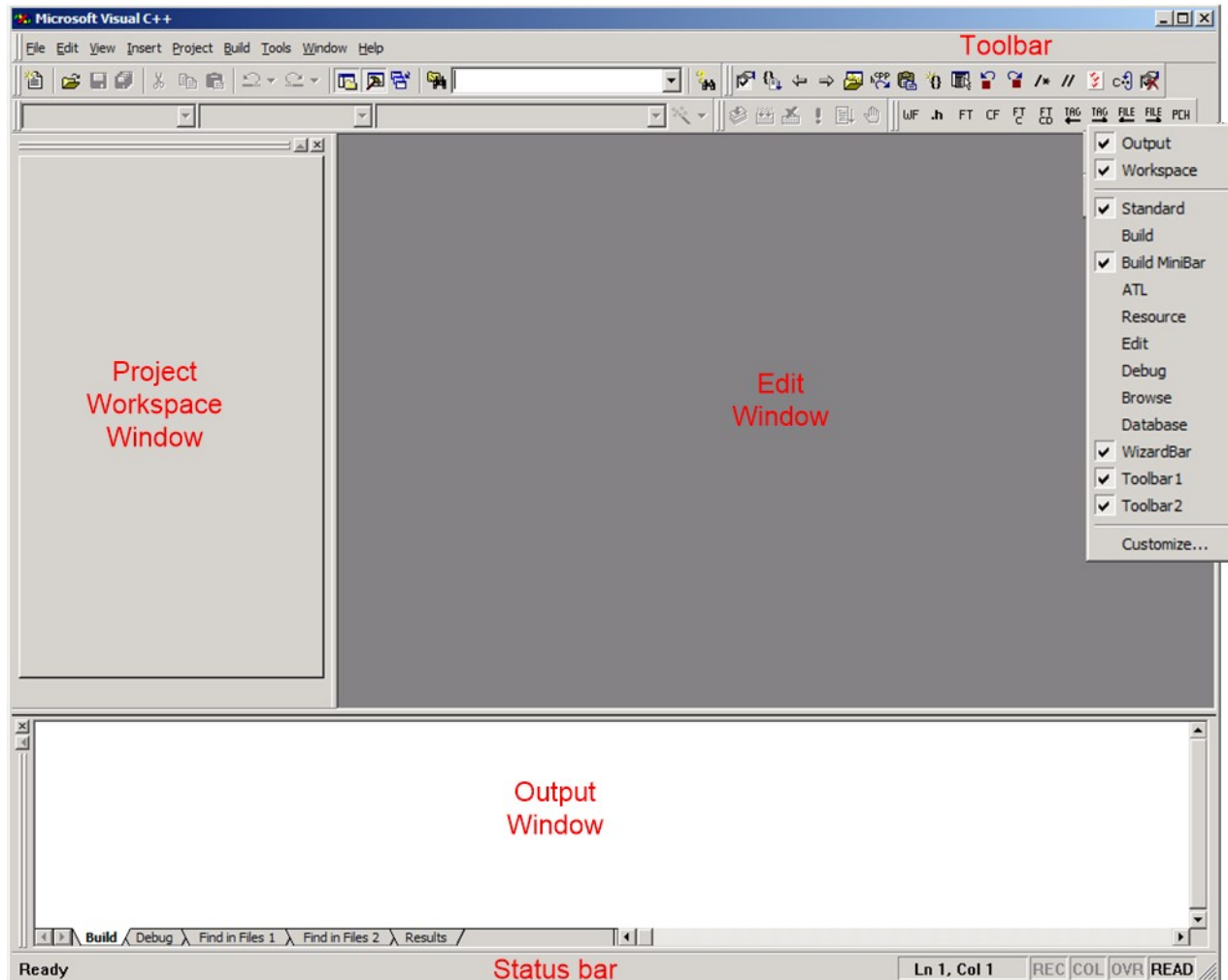


# Windows 프로그래밍의 기본 구조



# VC++ IDE (통합 개발 환경)

- Visual Studio 인터페이스



# VC++ IDE (통합 개발 환경)

---

- 시스템의 구성 요소
  - Editor
    - Source code 작성
    - Compile, editing, debugging 등의 interactive 환경 제공
  - Compiler
    - Compile : Source code → machine language
    - Error and warning detecting and notifying
    - Output : object code (.obj)
  - Linker
    - Link : module merging + program libraries
    - Linking error detecting and notifying
  - Library
    - Standard library
    - MFC & T (Microsoft Foundation Classes and Templates)
- 개발 과정 단순화를 위한 기타 도구
  - **AppWizard** : Windows 프로그램의 기본적인 Framework 자동 생성
  - **ClassWizard** : AppWizard에 의해 생성된 클래스 확장 용이성 제공

# 윈도우 응용 프로그램 개발 도구 (1/3)

---

- API (Application Programming Interface) 기반
  - 운영체제가 응용 프로그램을 위해 제공하는 각종 함수의 집합
  - C 언어 기반
- 장점
  - 세부적인 제어가 가능
  - 윈도우 운영체제가 제공하는 모든 기능을 사용 가능
  - 생성 코드의 크기가 작고 속도가 빠름
- 단점
  - 생산성이 낮음

\*SDK(Software Development Kit): 컴파일러, 개발툴, 소스코드, 헤더, 라이브러리, 도움말 등 윈도우 응용 프로그램 제작에 필요한 집합



# 윈도우 응용 프로그램 개발 도구 (2/3)

---

- RAD (Rapid Application Development) 기반
  - 시각적으로 화면을 디자인하고 코드를 추가하는 방법으로 프로그램을 빠르게 개발 가능
  - Visual Basic, Delphi 등
- 장점
  - 생산성이 높음
- 단점
  - 일반적으로 생성 코드의 크기가 크고 실행 속도가 느림
  - 운영체제가 제공하는 모든 기능을 활용한 세부적인 제어가 어려움

# 윈도우 응용 프로그램 개발 도구 (3/3)

---

- MFC (Microsoft Foundation Class Library) 기반
  - API 중 많이 사용되는 기능들을 사용하기 쉽게 잘 포장해 놓은 것
  - 객체 지향 언어를 이용하여 클래스로 제공
- 장점
  - API 직접 이용 보다 생산성이 높음
  - RAD보다 생성 코드의 크기가 작고 실행 속도가 빠름
- 단점
  - 초기 학습 시간이 긴 편
    - 객체 지향 언어
    - 클래스 라이브러리 구조와 기능 및 각 클래스의 관계 파악

Q & A