Lecture # 3

Microsoft Assemblies

Engr. Fiaz Khan

Today's Agenda

01 **Monolithic**

02 **Component-based**

03 Assemblies

04 **Global Assembly Cache**

05 **Creating a .NET Component**

06 Creating a .NET Component Containing Multiple Classes

اَلْحَمْدُ لِلّٰهِ رَبِّ الْعٰلَمِيْنَ وَ الصَّلٰوةُ وَ السَّلَامُ عَلٰى سَيِّدِ الْمُرْسَلِيْنَ

اَمَّا بَعْدُ فَاَعُوْذُ بِاللّٰهِ مِنَ الشَّيْطٰنِ الرَّجِيْمِ ۙ بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِيْمِ

اَللّٰهُمَّ افْتَحْ عَلَيْنَا حِكْمَتَكَ وَانْشُرْ عَلَيْنَا رَحْمَتَكَ يَا ذَا الْجَلَالِ وَالْاِكْرَام

Translation

O Allah عَزَّوَجَلَّ! Open the door of knowledge and wisdom for us, and have mercy on us! O the One Who is the Most Honourable and Glorious! *(Al-Mustatraf, vol. 1, pp. 40)*
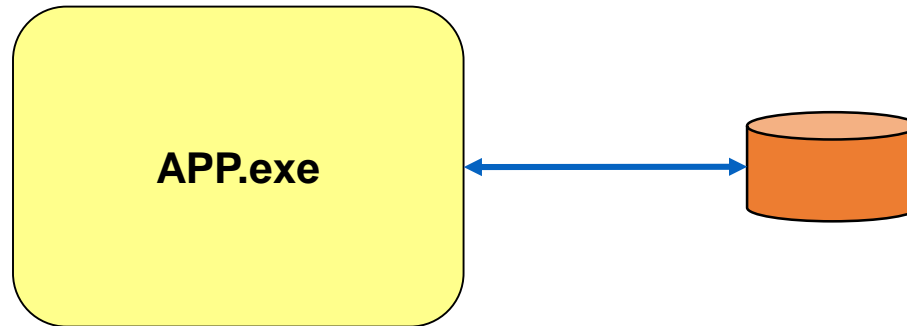
Lets Start Learning

Whoever recites Salat upon me one time, Allah عَزَّوَجَلَّ sends ten blessings upon him. *(Sahih Muslim, pp. 216, Hadees 408)*

جس نے مجھ پر ایک بار دُرُودِ پاک پڑھا اللہ عَزَّوَجَلَّ اُس پر دس رَحْمتیں بھیجتا ہے۔
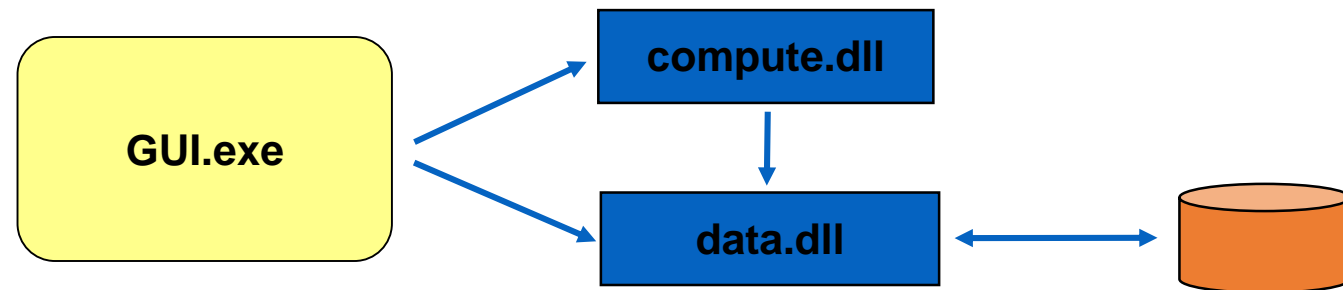
(مُسلِم ص۲۱۶ حدیث۴۰۸)

Monolithic

- Monolithic app: all source code compiled into one .EXE

- ***not*** the norm on Windows…

Component-based

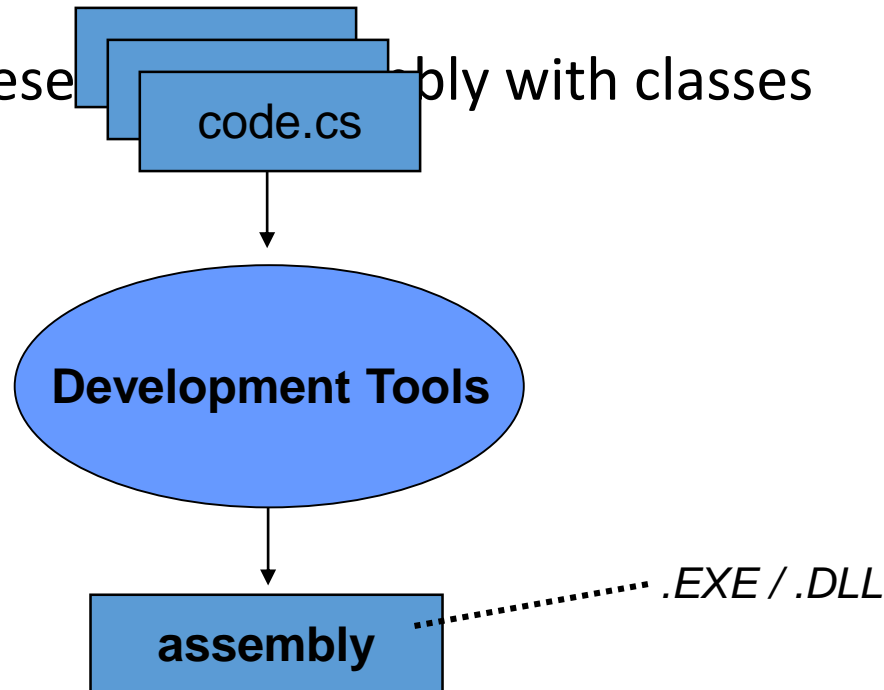- Component-based app: .EXE + 1 or more .DLLs

- standard practice on Windows…

Why component-based?

- Many motivations:
  - team programming
  - multi-language development (I like VB, you like C#)
  - code reuse (e.g. across different .EXEs)
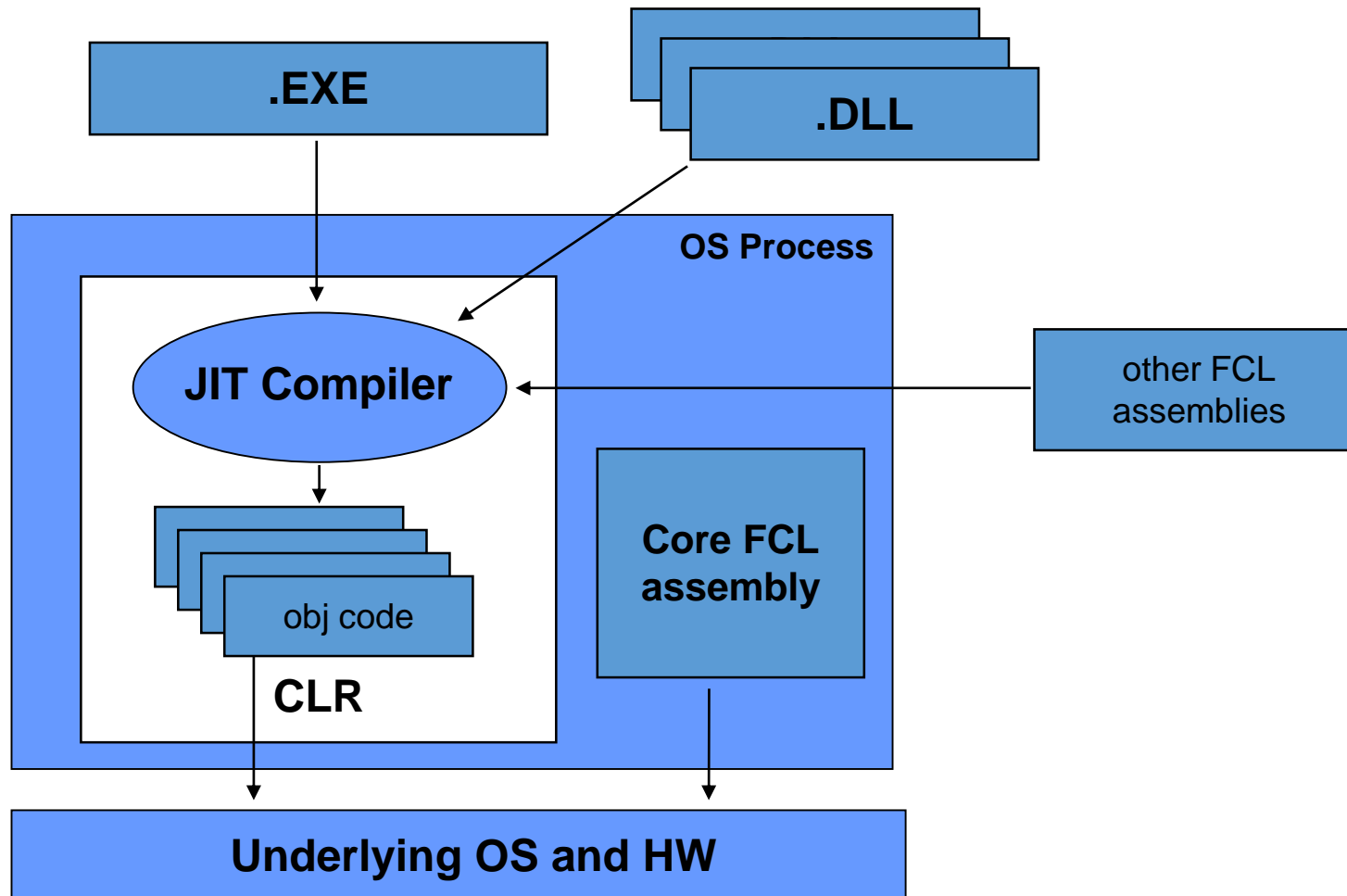 - independent updating (update just component X)

Assemblies

- .NET packages components into *assemblies*

- 1 assembly = 1 or more compiled classes

  - .EXE represents an assembly with classes + Main program

  - .DLL represe~~~~bly with classes

code.cs

Development Tools

assembly .... *.EXE / .DLL*
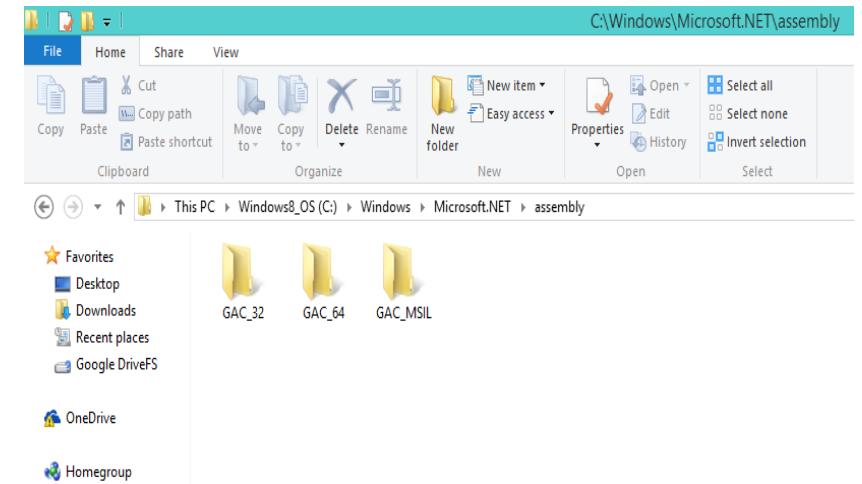
CLR-based execution revisited

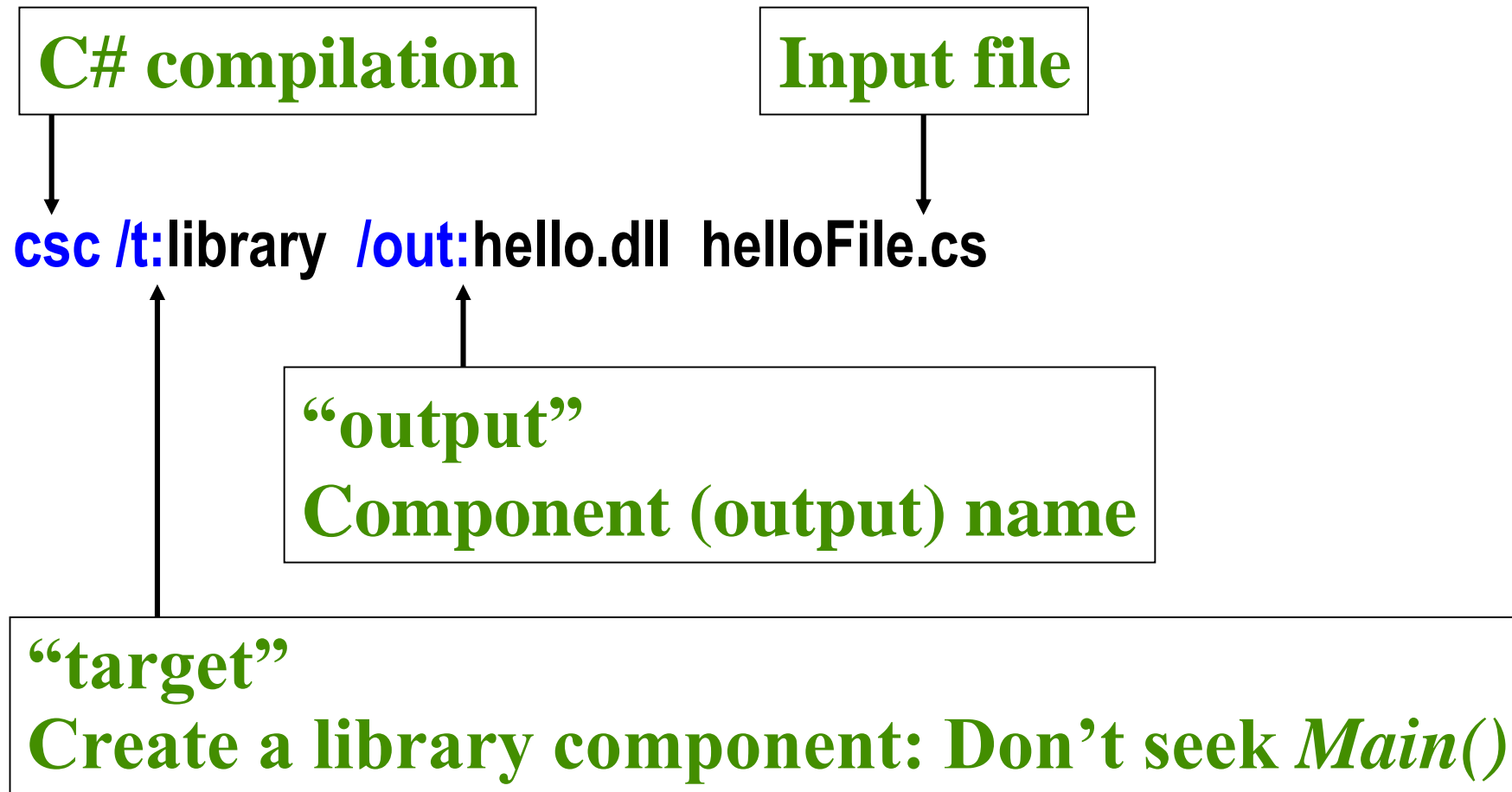- CLR must be able to locate all assemblies:

GAC?

- GAC = Global Assembly Cache
  - C:\Windows\Microsoft.NET\assembly

- Let's break it down:
  - global - applies to the entire machine
  - assembly - what .NET calls its code-libraries (DLLs)
  - cache - a place to store things for faster/common access

  So the GAC must be a place to store code libraries so they're accessible to all applications running on the machine.
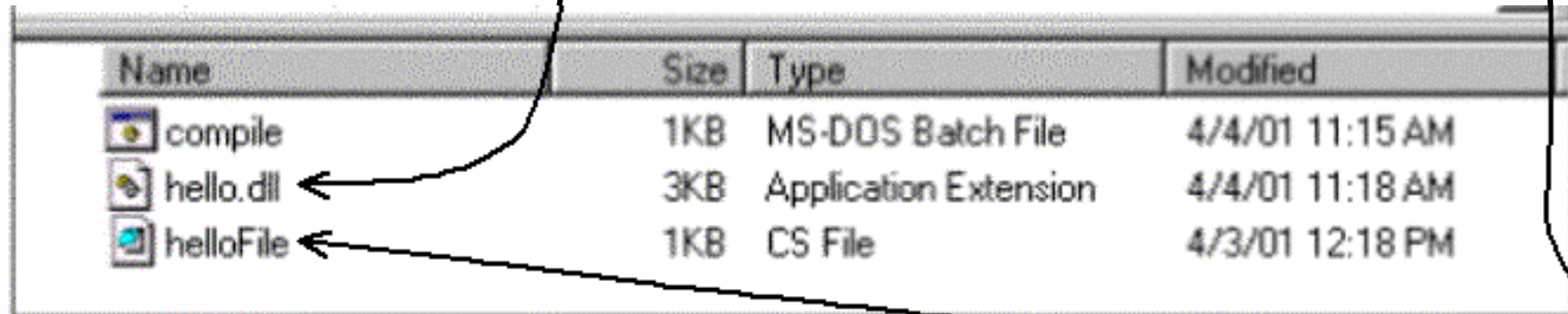
Creating a .NET Component *1 of 2*

C# compilation

Input file

csc /t:library /out:hello.dll helloFile.cs

"output"
Component (output) name

"target"
Create a library component: Don't seek *Main()*

helloFile.cs

csc /t:library /out:hello.dll helloFile.cs

hello.dll

Name	Size	Type	Modified
compile	1KB	MS-DOS Batch File	4/4/01 11:15 AM
hello.dll	3KB	Application Extension	4/4/01 11:18 AM
helloFile	1KB	CS File	4/3/01 12:18 PM

1

Using a .NET Component

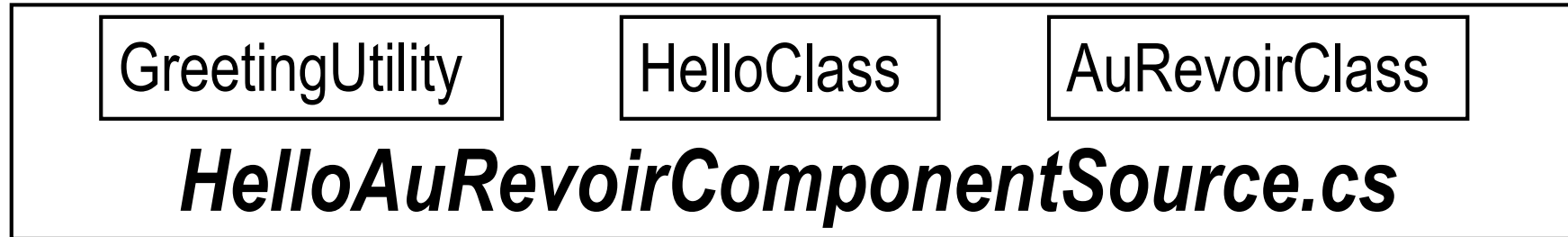**helloClient.cs**   **hello.dll**

csc /r:hello.dll /out:helloApp.exe helloClient.cs

helloApp.exe

… now execute helloApp.

Creating a .NET Component Containing Multiple Classes

| GreetingUtility | HelloClass | AuRevoirClass |

HelloAuRevoirComponentSource.cs

csc /t:library /out:helloAuRevoir.dll

HelloAuRevoirComponentSource.cs

helloAuRevoir.dll

Output For "Multiple Class" Example


```
greetingApp

Auto

Client use of a component containing an internal class:
Component says: Hello World
Component says: Au Revoir World
_
```

ENGR. FIAZ KHAN

LECTURER COMPUTER SCIENCE

TELF/EF SET / ACEPT CERTIFIED

MICROSOFT ACADEMY TRAINER

ORACLE CERTIFIED PROFESSIONAL

GOOGLE CERTIFIED PROFESSIONAL

Lahore, Pakistan

+92-336-678-2755

muhammad.fiaz@pucit.edu.pk

https://www.linkedin/in/fiazofficials

THANK YOU!

Do you have any questions?