.NET PLATFORM

Asma Kanwal
Lecturer
Department of Computer Science
GC University, Lahore

.Net Solution

Characteristics:

- Comprehensive interoperability with existing code
- Complete and total language integration
- A common runtime engine shared by all .NET-aware languages
- A comprehensive base class library
- A truly simplified deployment model

Building Blocks of .Net Platform

- **CLR-** Common Language Runtime Engine
- **CTS-** Common Type System
- **CLS-** Common Language Specification

CLR

Role:

- Locate , load, and manage
- Memory management
- Creating application domains
- Threads
- Object context boundaries
- Performing various security checks

CTS

Role:

- Fully describes all possible data types and programming constructs supported by the runtime.
- Specifies how these entities can interact with each other.
- Details how they are represented in the .NET metadata format.

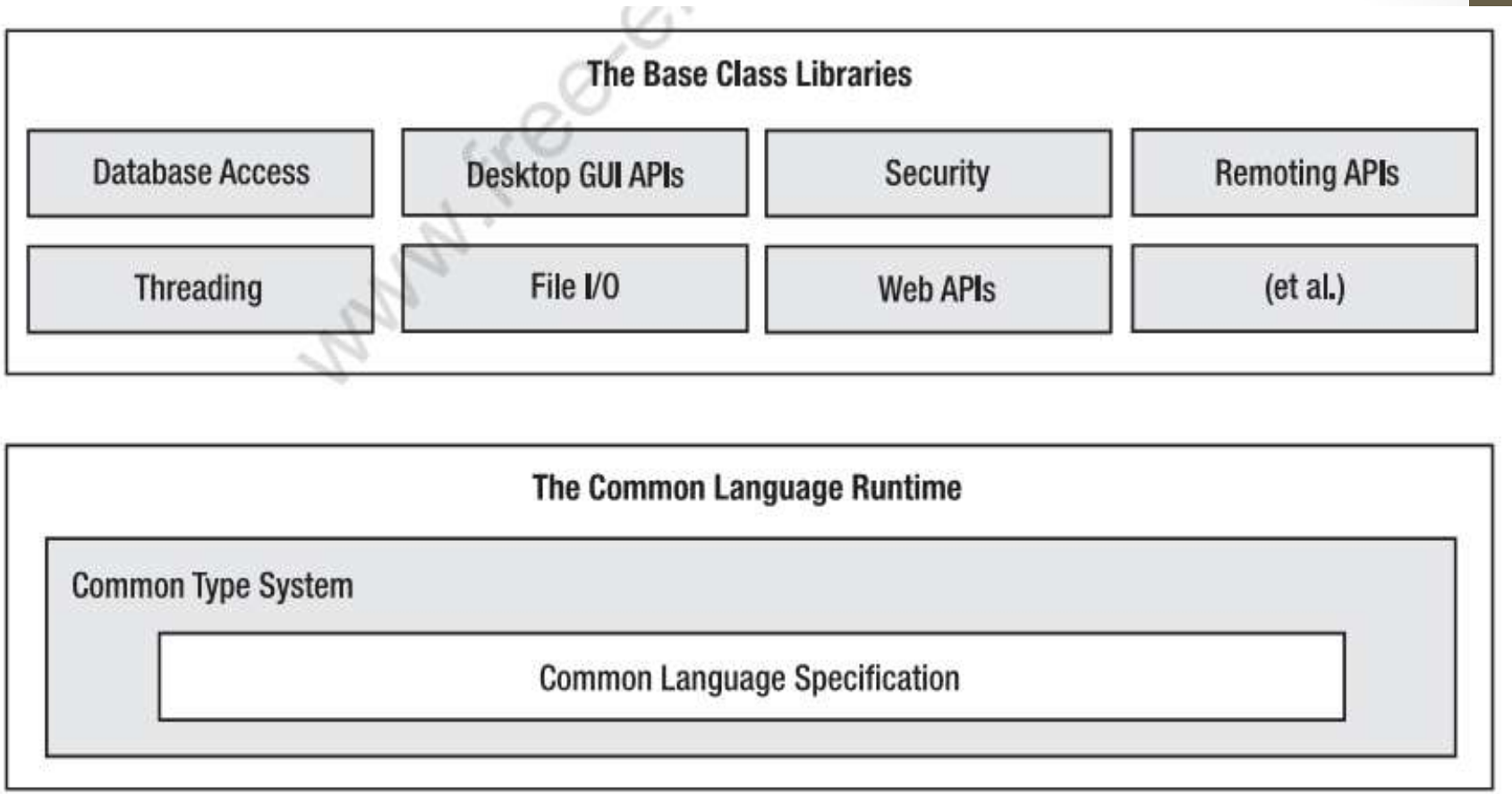
CLS

Role:

Defines a subset of common types and programming constructs.

.NET types that only expose CLS-compliant features, you can rest assured that all .NET-aware languages can consume them. Conversely, if you make use of a data type or programming construct that is outside of the bounds of the CLS, you cannot guarantee that every .NET programming language can interact with your .NET code library

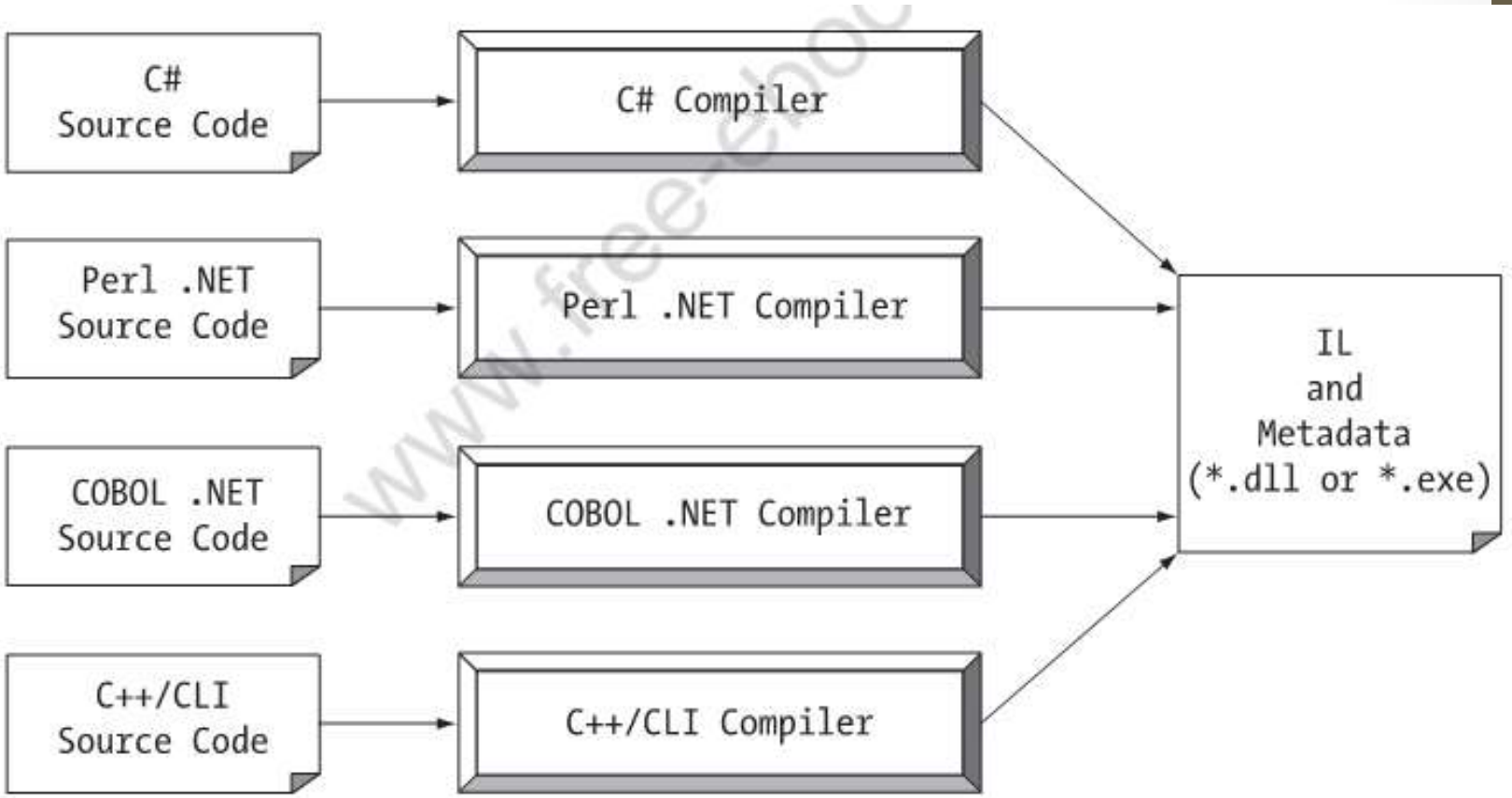
Base Class Libraries



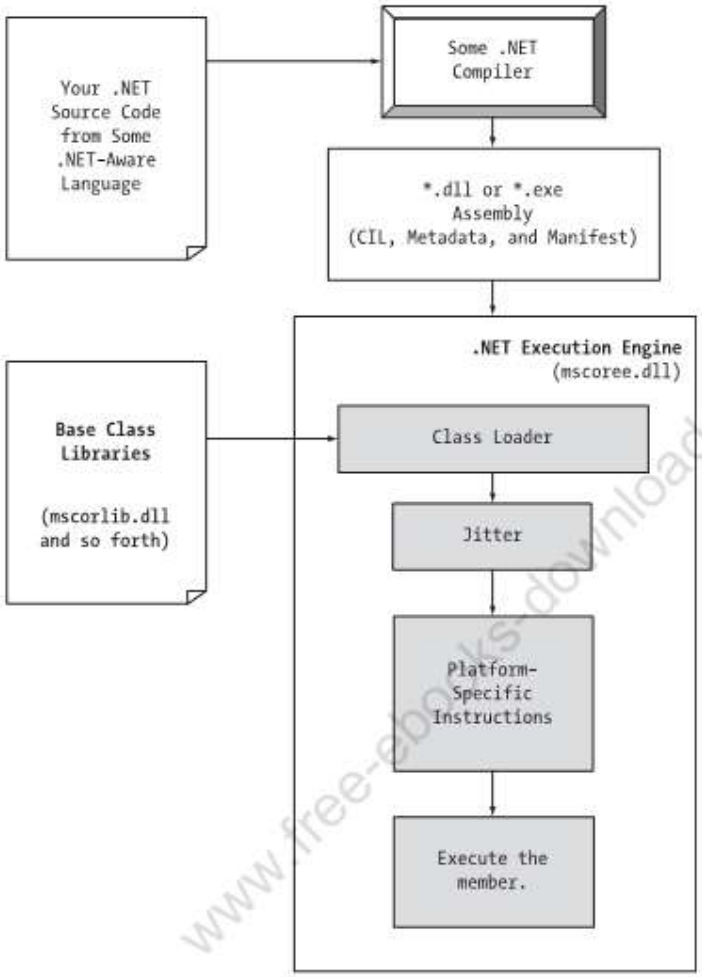
C# Characteristics

- No pointers required!
- Automatic memory management through garbage collection
- Formal syntactic constructs for classes, interfaces, structures, enumerations, and delegates
- The C++-like ability to overload operators for a custom type, without the complexity
- Support for attribute-based programming.

.NET Assemblies



CLR Working


Your .NET Source Code from Some .NET-Aware Language
Some .NET Compiler
*.dll or *.exe Assembly (CIL, Metadata, and Manifest)
.NET Execution Engine (mscoree.dll)
Base Class Libraries
(mscorlib.dll and so forth)
Class Loader
Jitter
Platform-Specific Instructions
Execute the member.

Compilation and Execution