Managed Languages

Syllabus

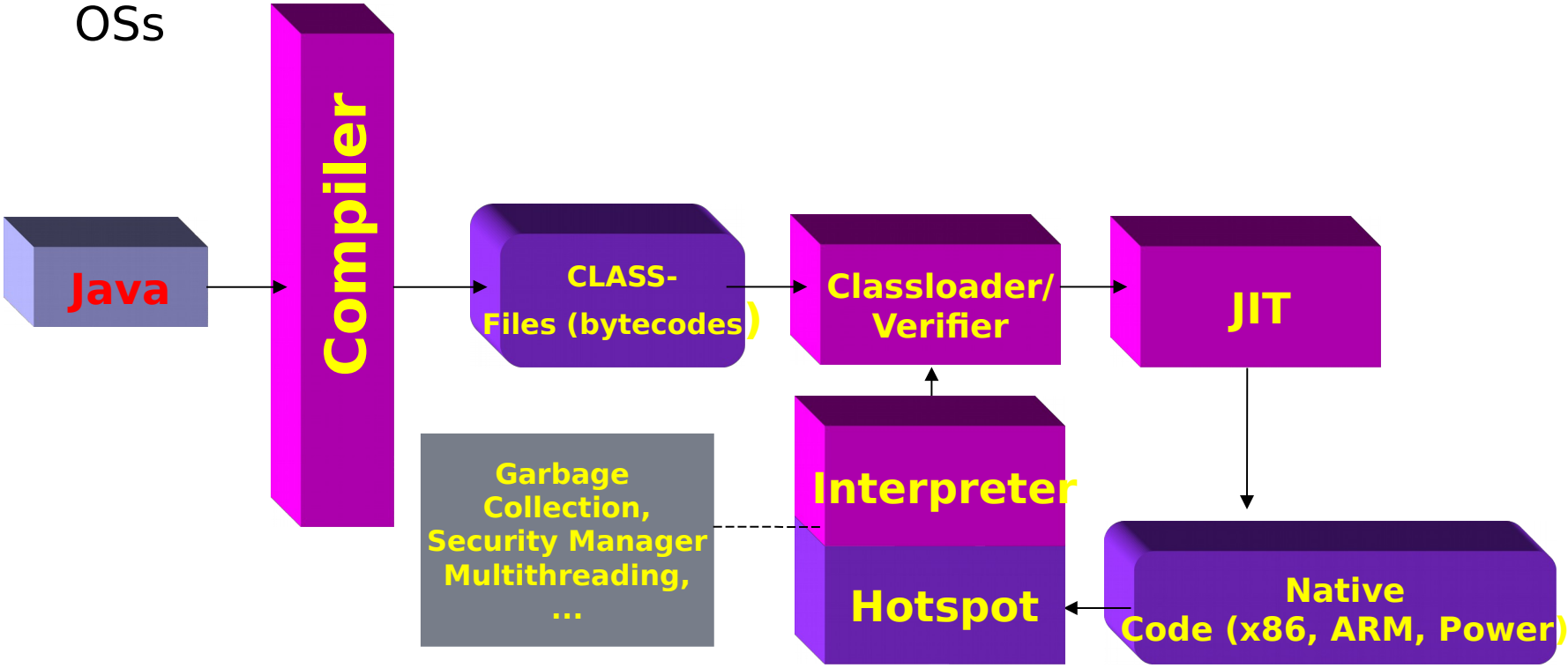
Lecture Series (hours)	Topics
1-4	Introduction and Motivation, Paradigms
5-10	Syntax and Semantics, BNF, Compilation
11-18	Data Types, Constructs, Functions, Activation Records, Names and Bindings
19-28	Concurrency, Lambda Calculus, Functional PLs, Logical PLs, Event driven programming
29-36	Virtual Machines, Managed Languages, JIT, Case study

Managed Languages

- Has its own runtime
- Usually governed legally to a company
- Has a built-in compiler, runtime, and usually platform/language independent
- Two popular examples: Java and .NET frameworks
- Case study: Java
- Compare with .NET

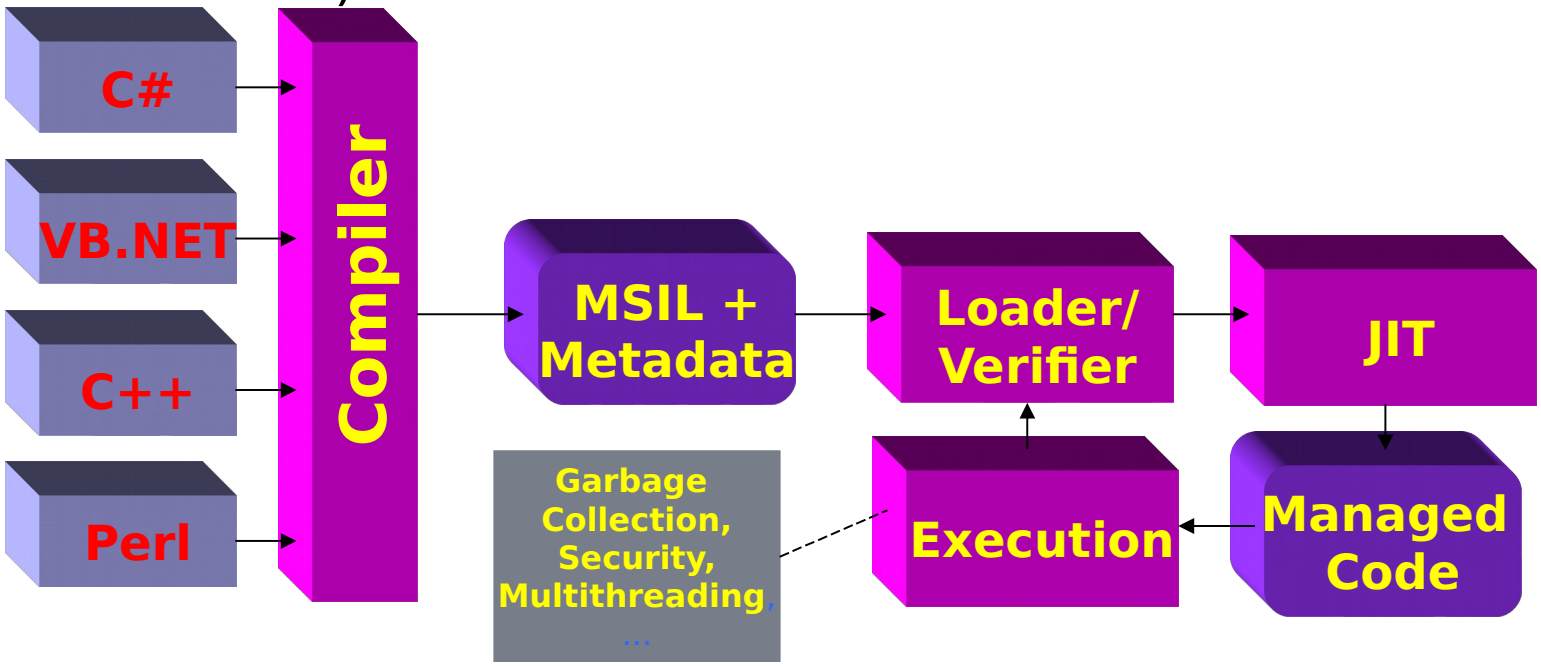
Java Virtual Machine

- The JVM is intended for Java and interprets Java Byte Code.
- Other languages can be compiled to Java bytecode, however (e.g. Ada)
- Just-in-Time compilers exist for different environments and OSs



.NET Runtime

- It is called the Common Language Runtime (CLR)
- It is intended for any language compiled to the MSIL
- Provides integration for several languages
- Provides support for non-OO languages (e.g. tail recursion)



Comparison - runtimes

- J2SE (Standard) runtime
  - desktop applications
- J2EE (Enterprise) runtime
  - web applications
- J2ME (Micro) runtime
  - runtime for gadgets

- C# or VB.NET with VS Express
  - desktop application
- IIS with VS Express Web
  - web applications
- .NET compact framework
  - runtime for gadgets

Java platform - .NET platform

- Java Virtual Machine (JVM)

aka Java Runtime Environment (JRE)

  - Linux, Windows, Mac and Unix
  - download from Sun
    - JIT compiler and libraries

- Java Development Kit (JDK)
  - Java compiler and utilities
    - Java bytecode

- integrated development env. (IDE)
  - Eclipse (free - IBM)
  - Netbeans (free - Sun)

FREE

- application servers
  - Tomcat
  - Glassfish (Sun)
  - BEA Weblogic
  - IBM Websphere

- Microsoft .NET Framework 2.0

aka .NET common language runtime (CLR)

  - all versions of Windows (40+)
  - download from Microsoft
    - JIT compiler and libraries

- NET framework 2.0 SDK
  - C# and VB.NET compilers and utilities
    - MS Intermediate Language (MSIL)

- integrated development env. (IDE)
  - Visual Studio Express (free - Microsoft)

NOT

- application servers
  - Microsoft Internet Information Server (IIS)

features 1

- virtual machine
  - platforms (all major OS's)
  - spec
  - implementations
- libraries
- languages
  - Java
  - Jython
  - Groovy
- web servers (many vendors)
  - platforms (Unix, Linux)
  - scalability
  - cost
- web capabilities
  - servlet
  - JSP
  - JSF

- virtual machine
  - platforms (all versions of Windows)
  - spec
  - implementations
- libraries
- languages
  - C#, VB.NET, J# from Microsoft
  - many others from third parties

  (Haskell, Lisp, Python, COBOL, Fortran, etc.)
- web servers (just one!)
  - platforms (most Windows)
  - scalability
  - cost
- web capabilities
  - handler
  - ASP (.NET)
  - (forgot equivalent name)

features 2

- native code calling
- components
  - beans
- environments
  - applet (in browser)
  - servlet (in server)
  - Web Start
    - –installs from web
    - –caches on user's PC
- deployment
  - .jar
  - .war
  - .ear
  - .class

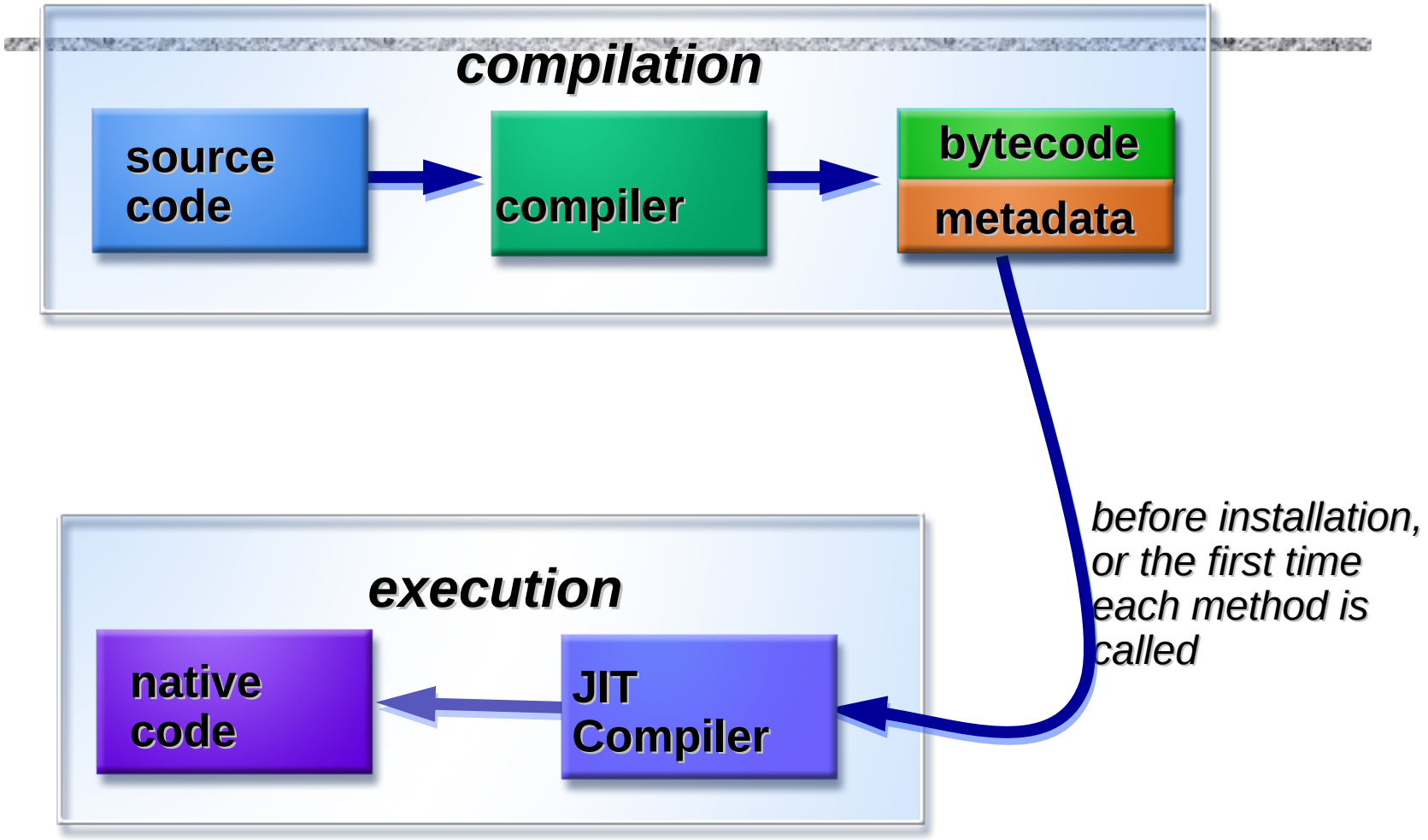
- native code calling
- components
  - .DLL
- environments
  - ActiveX (in browser)
  - handler (in server)
  - Smart Client
    - –installs from web
    - –caches on user's PC
- deployment
  - .exe (on file system)
  - .exe (in GAC)
  - .dll (on file system)

features 3

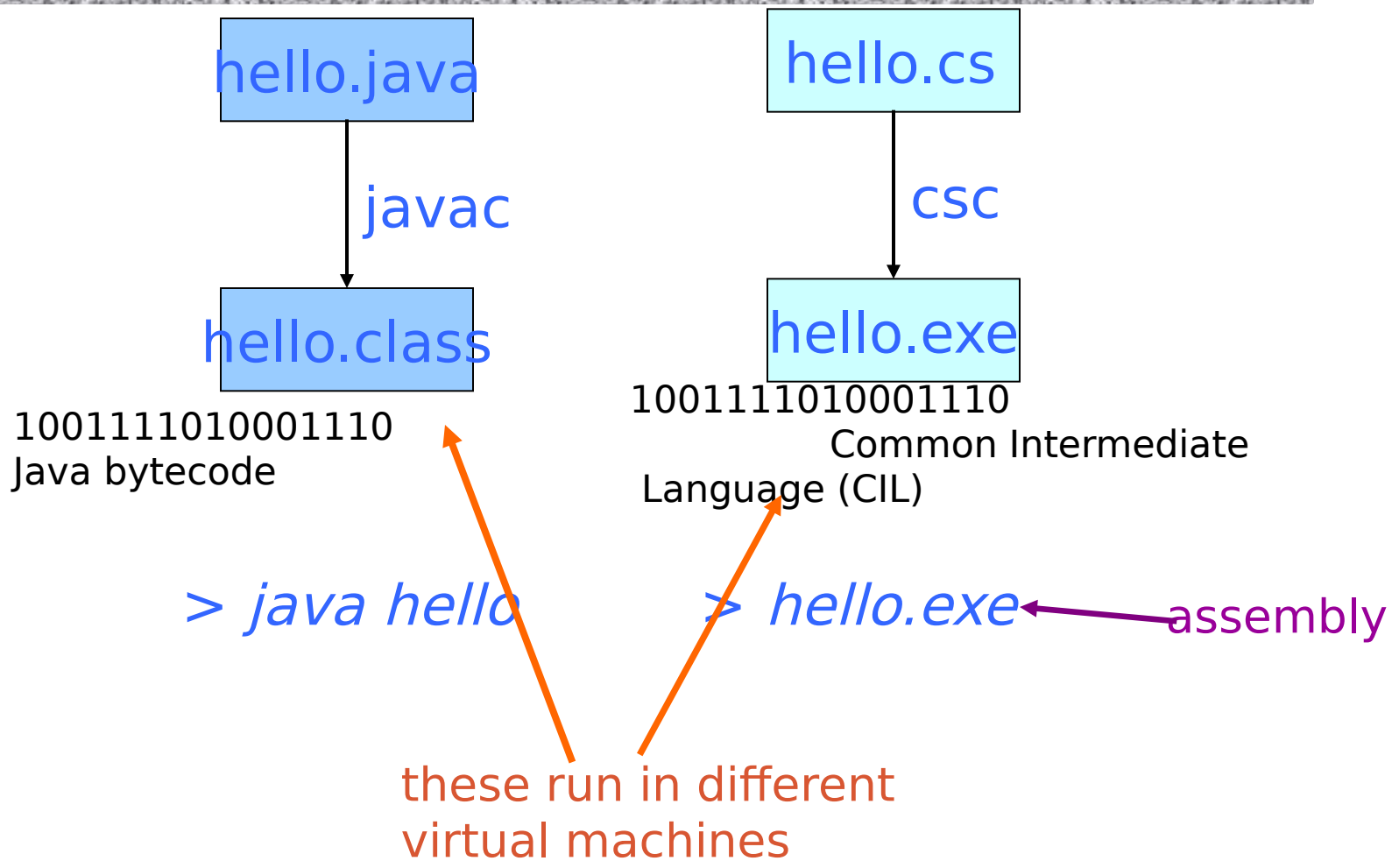
- databases
  - JDBC
- CORBA
  - binary object remoting
- XML
  - via 3rd-party add-ons until Java 6
- IDE's
  - Eclipse, with 1000's of plugins
  - NetBeans (from Sun) – also free
- service oriented architecture (SOA)
  - annotations appearing
  - web services WS-I

- databases
  - ODBC
- COM
  - binary object remoting
- XML
  - excellent support early one
- IDE's
  - free versions of Visual Studio
  - some third party IDE's
- service oriented architecture (SOA)
  - annotations
  - web services WS-I

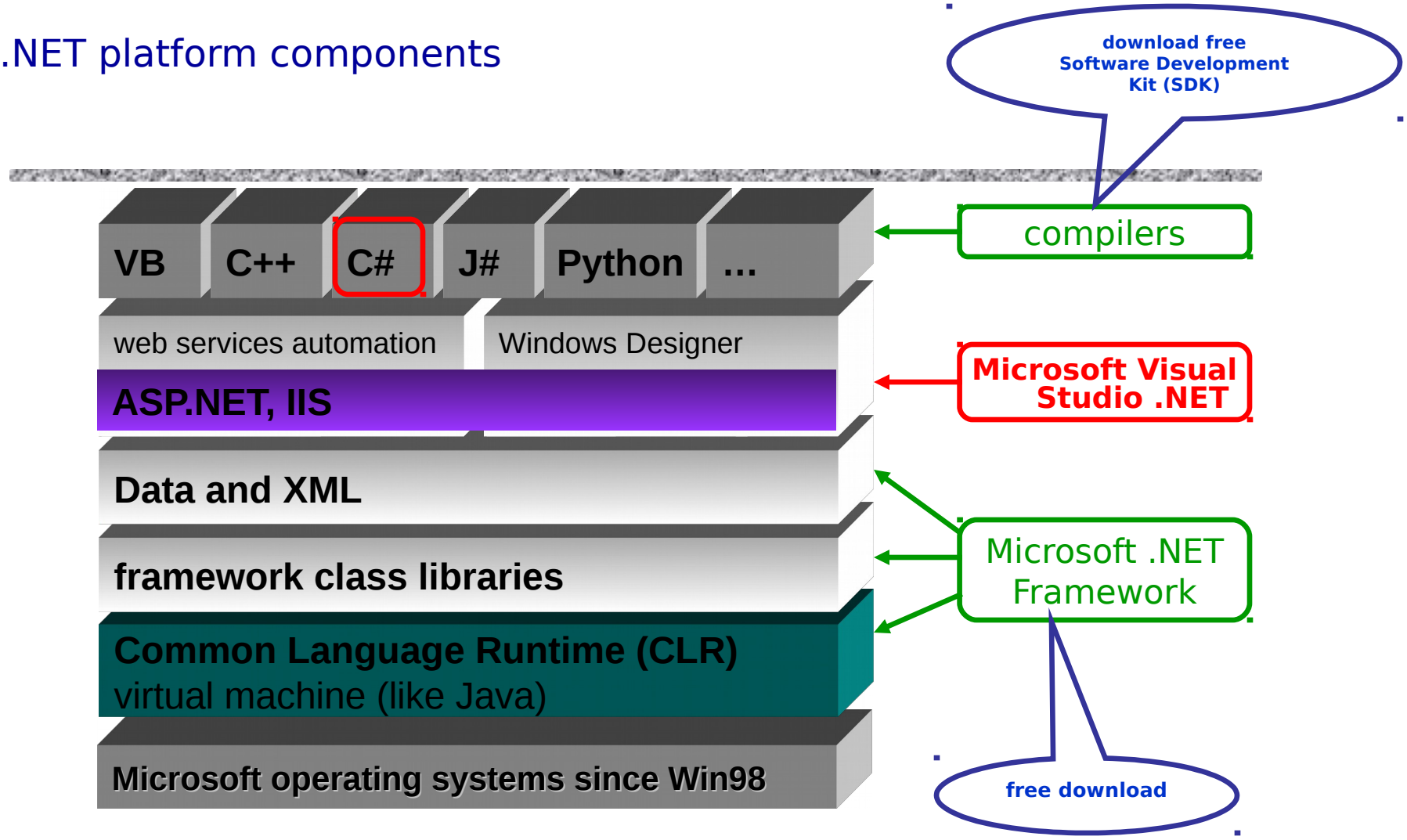
Just in Time (JIT) compilers in JRE (JVM) and .NET runtimes



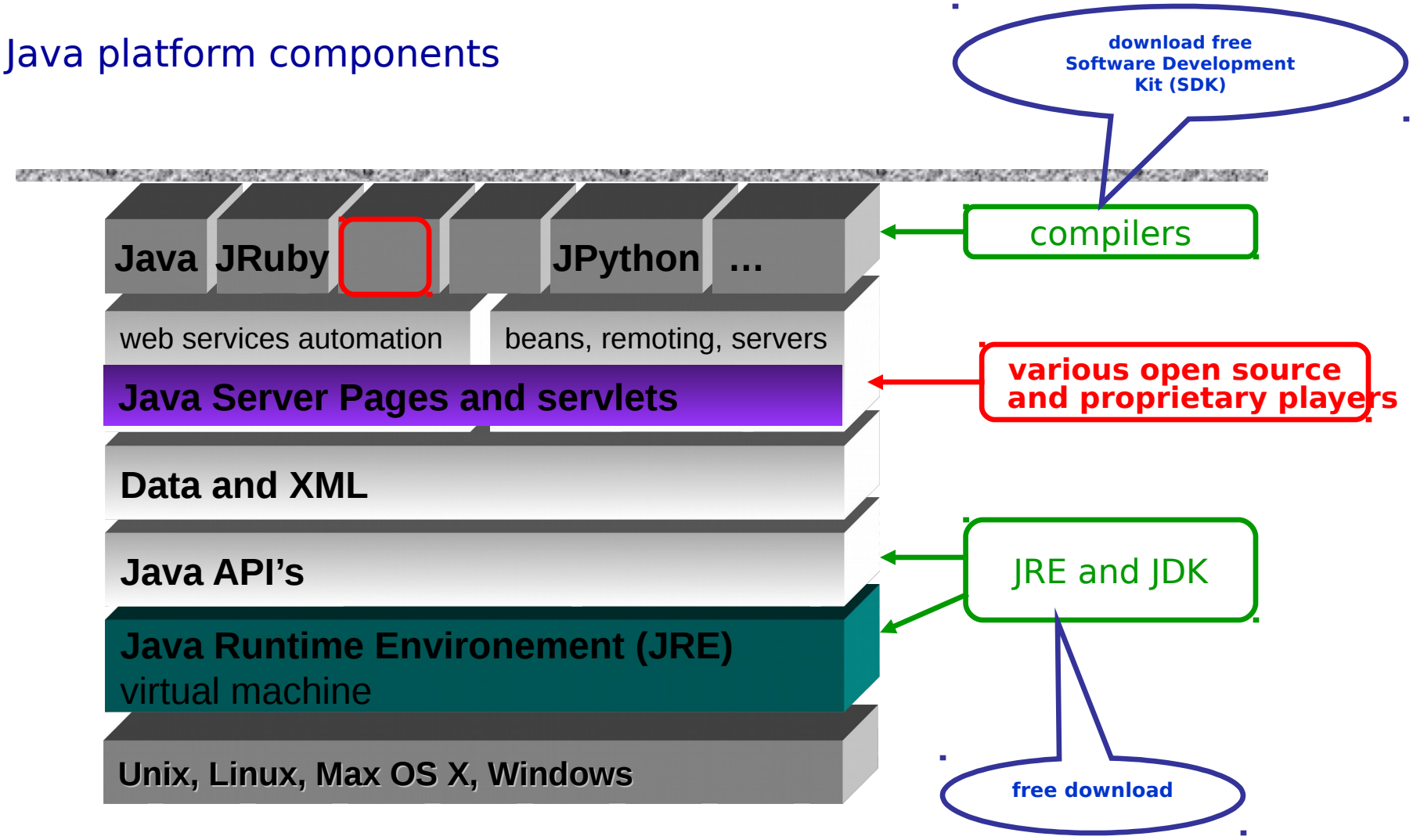
console commands for compiling Java and C#.NET



.NET platform components



Java platform components



who implements Java runtimes?

Sun Microsystems

- Java HotSpot Virtual Machine for Windows, Linux, Unix

Hewlett-Packard

- Java runtime for HP-UX, OpenVMS, Tru64, Reliant(Tandem) UNIX)

IBM

- Java runtime for MVS, AIX, OS/400, z/OS

Apple Computer

- MacOS Runtime for Java (MRJ)
  - J2SE built-in on Mac OS X
  - includes JDK (compilers)

BEA Systems

- JRockit (for their web server)

Syllabus

Lecture Series (hours)	Topics
1-4	Introduction and Motivation, Paradigms
5-10	Syntax and Semantics, BNF, Compilation
11-18	Data Types, Constructs, Functions, Activation Records, Names and Bindings
19-28	Concurrency, Lambda Calculus, Functional PLs, Logical PLs, Event driven programming
29-36 (>40)	Virtual Machines, Managed Languages, JIT, Case study (Java)

Some final notes

- End-Sem will focus more on second half of the course – will be for 50 marks (from activation records onwards)
- End-Sem on April 23, 2018 (Monday) – 3 hours
- Will be similar to Mid-Sem
- Will be available on April 28, Sat, 2pm-4pm, for you to check your answer scripts for both Mid/End Sem papers and any FINAL clarifications
- Best of luck with your exams and your course!
- My email id: srp1970@gmail.com