

2005 2



2005 9

@ Lesson 1 –

,



[Redacted line of text]

[Redacted line of text]

[Redacted line of text]

[Redacted line of text]

[Redacted line of text]

@

1

Green Project 1991

Oak

**Java 1995.5**

December '90 by **Patrick Naughton**,
Mike Sheridan and **James Gosling**

Embedded in various
consumer electronic device

1992. 9. 3 Star 7

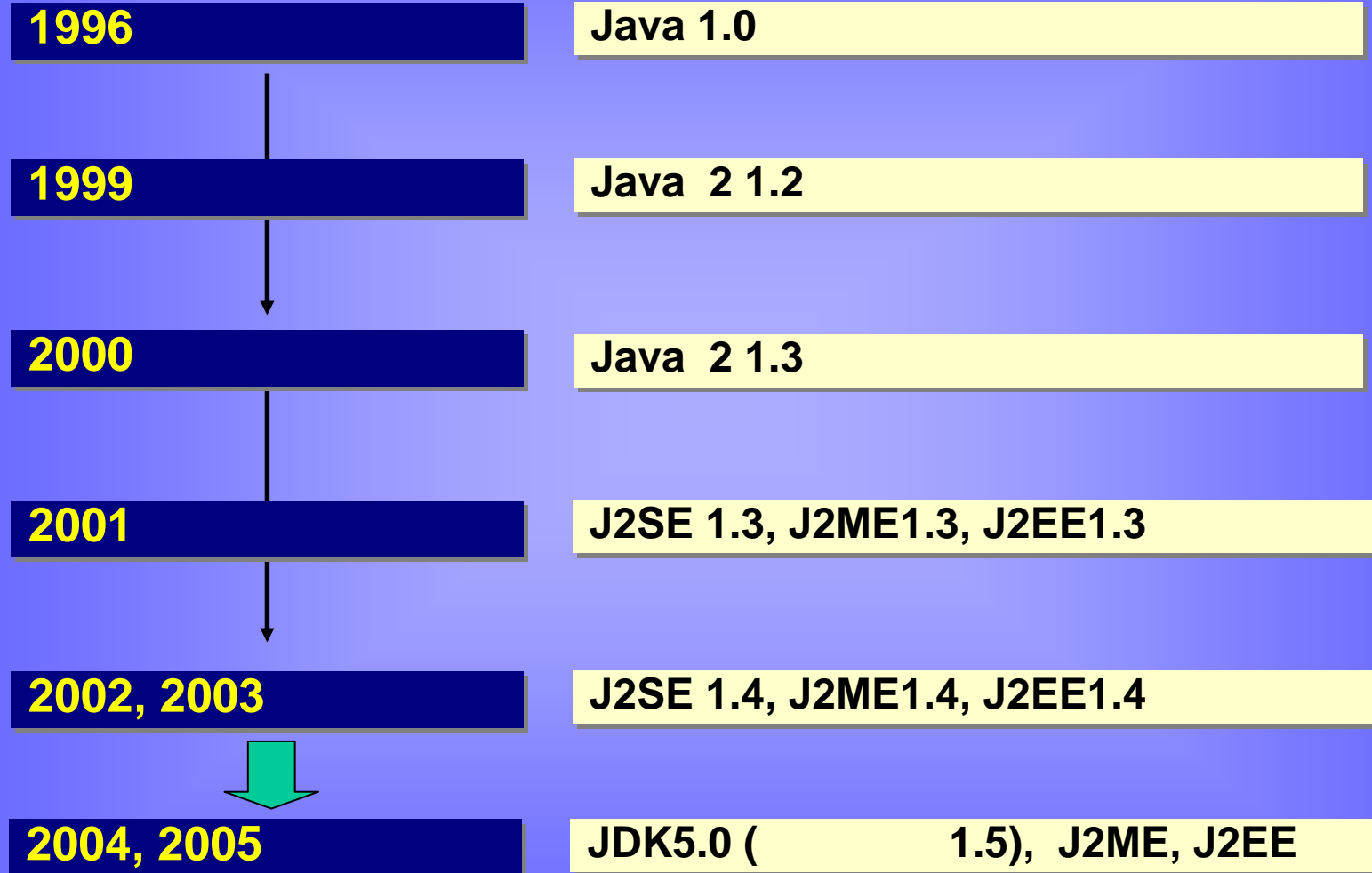
1993 www portability
cross platform- independent
1994 java

James Gosling, **Arthur Van Hoff**,
Andy Bechtolsheim

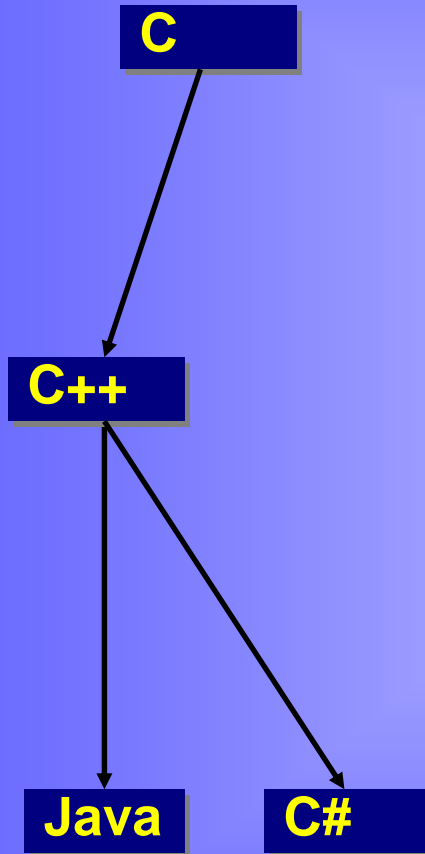
Hot java, java applet

@

2



@



Architectural-neutral, Portable:

. 가 .

Dynamic(Lazy loading):

,

Distributed :

.

Interpreted : Byte code

.

Multithread :

OO:

.

Robust:

.

Simple :

.

@

1

path

%JAVA_HOME%\bin

javac

java

javadoc

appletviewer

Java API**classpath**

-;

Java core API

rt.jar

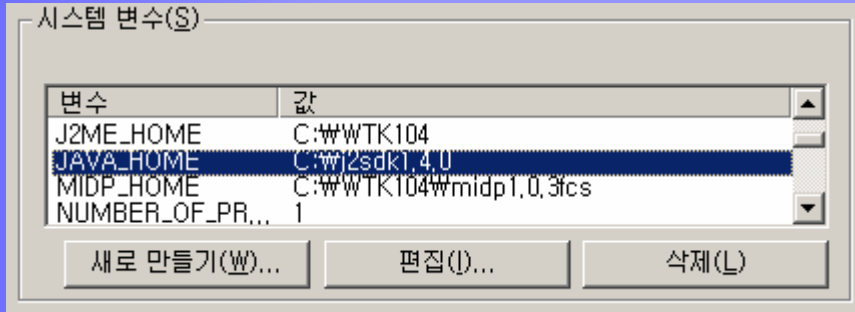
tools.jar

API

extension: servlet.jar,...

@

2



JAVA_HOME

C:

PATH

%JAVA_HOME%\bin;

CLASSPATH

.;

%JAVA_HOME%\lib\tools.jar

%JAVA_HOME%\jre\lib\rt.jar

%JAVA_HOME%\jre\lib\ext

@

(,)

javac

```
C:\W>javac
Usage: javac <options> <source files>
where possible options include:
  -g                Generate all debugging info
  -g:none           Generate no debugging info
  -g:<lines,vars,source>  Generate only some debugging info
  -O                Optimize; may hinder debugging or e
```

java

```
C:\#>java
Usage: java [-options] class [args...]
        (to execute a class)
or java -jar [-options] jarfile [args...]
        (to execute a jar file)
```

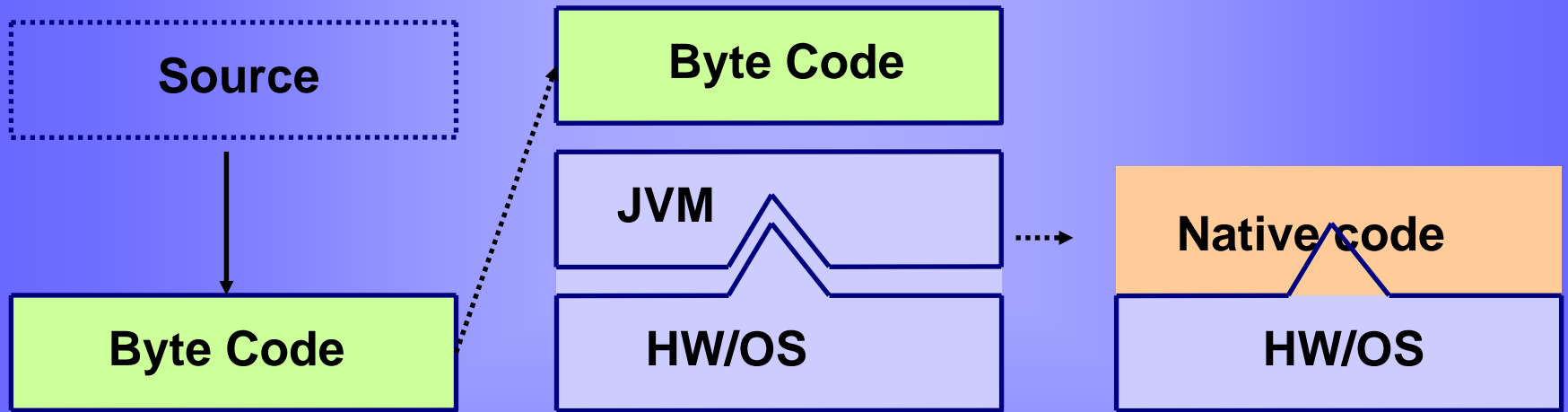
javap

```
Usage: javap <options> <classes>...
where options include:
  -b                Backward compatibility with javap in JDK 1.1
  -c                Disassemble the code
  -classpath <pathlist>  Specify where to find user class files
  -extdirs <dirs>      Override location of installed extensions
  -help            Print this usage message
  -J<flag>         Pass <flag> directly to the runtime system
  -l                Print line number and local variable tables
```


@

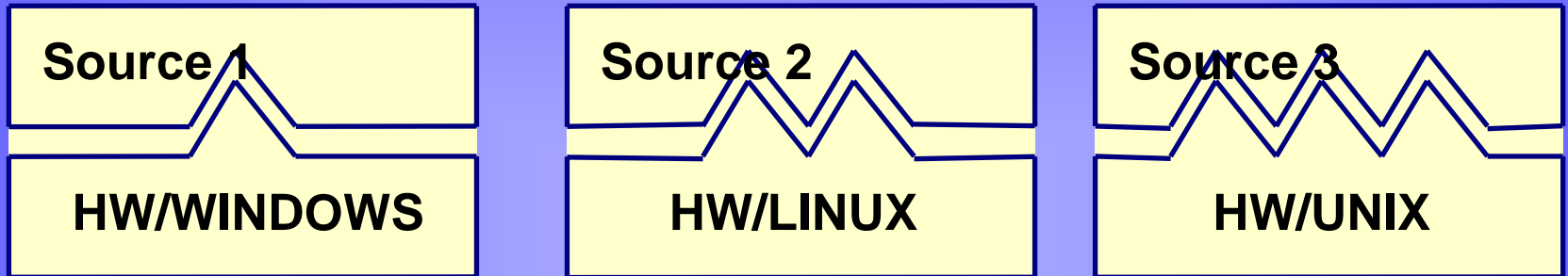
1

Virtual → Not Real Machine → 가

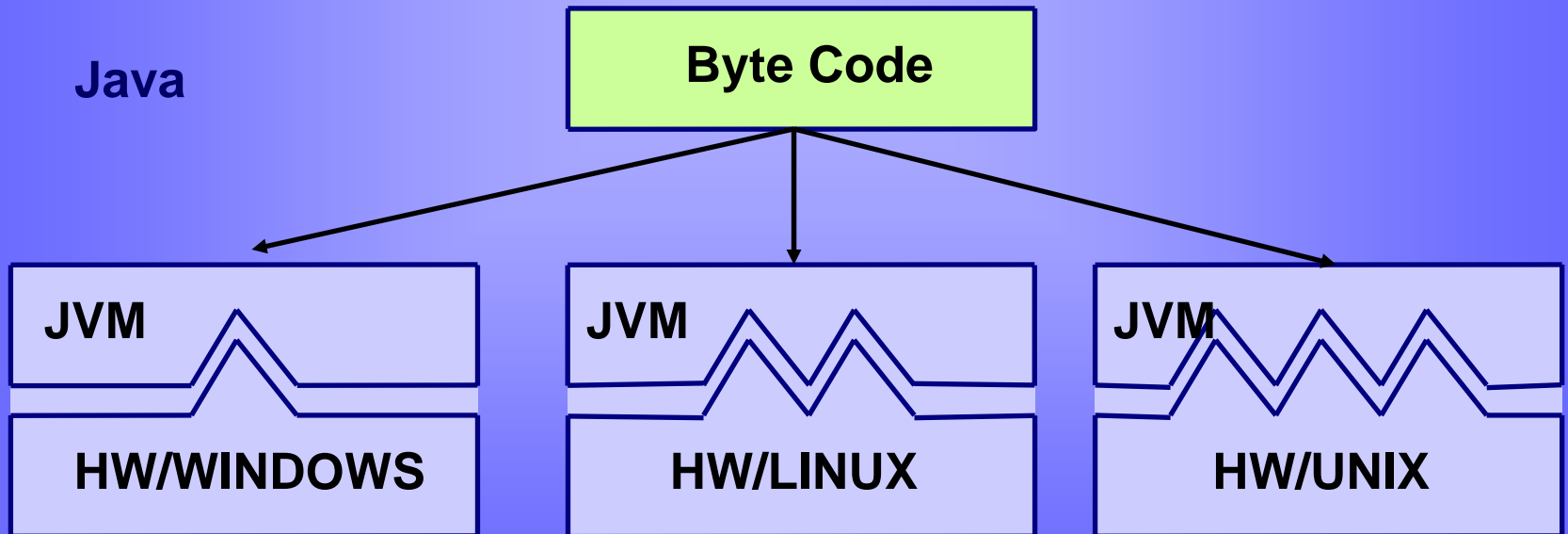


@ Legacy vs. Java

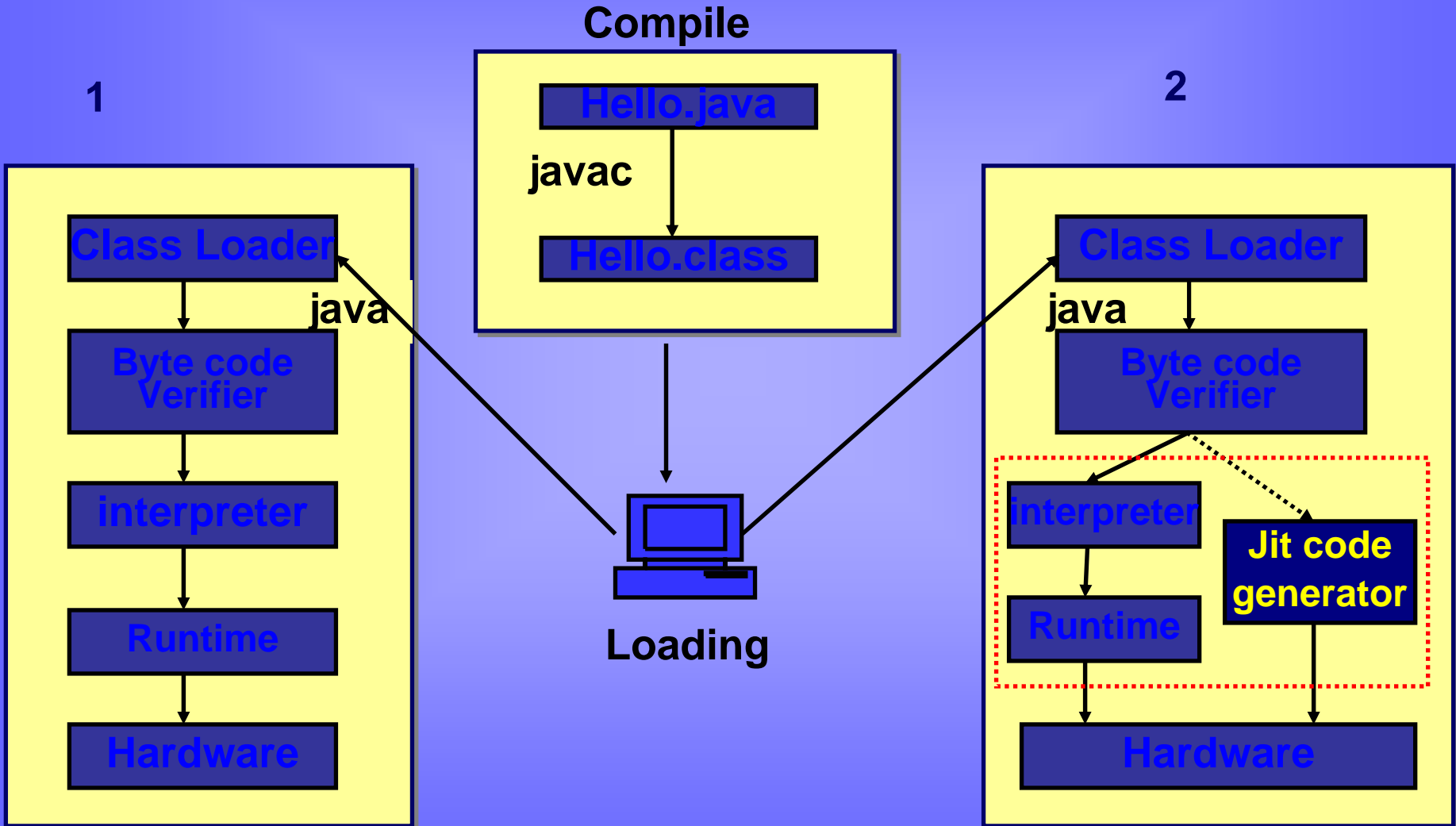
Legacy



Java



@ Compile Runtime



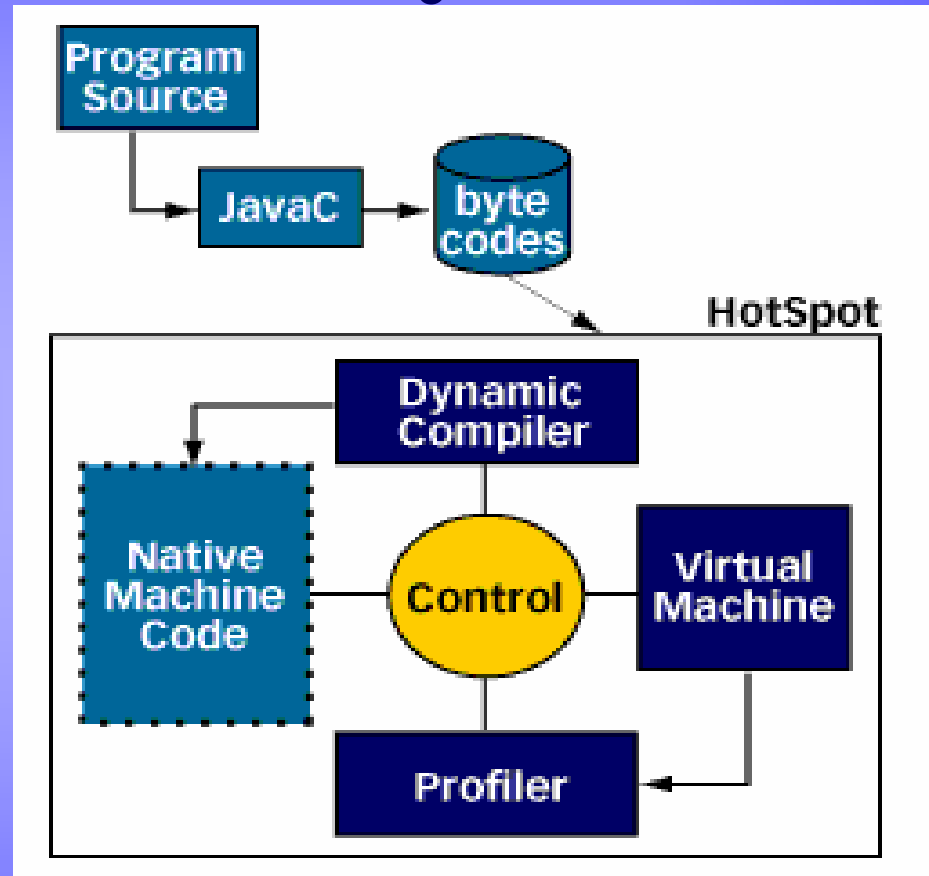
@ HotSpot & JIT

Just-in-time (JIT) compilers promise to improve the performance of Java applications.

Rather than letting the JVM run bytecode, a JIT compiler translates code into the host machine's native language.

Thus, applications gain the performance enhancement of compiled code while maintaining Java's portability.

3



@ Java2 Platform 1(Edition)



Midlet

Application

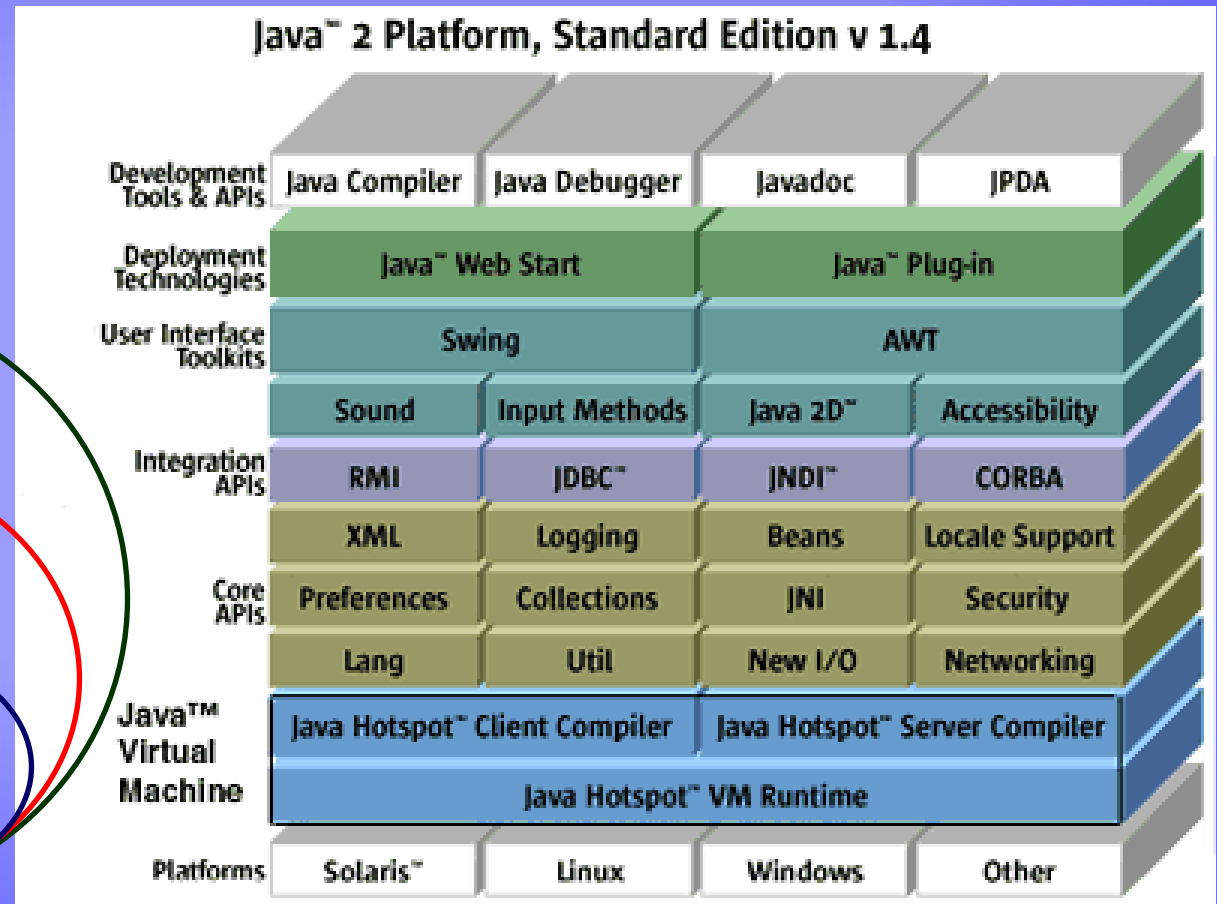
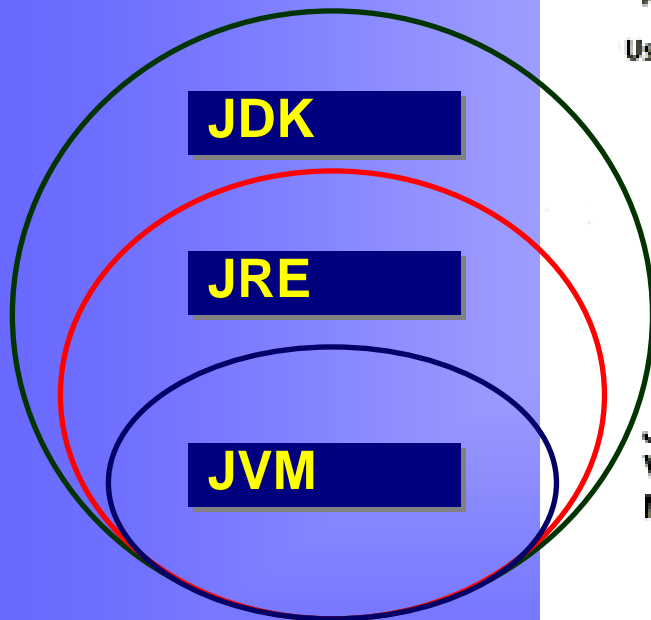
EJB

Applet

JSP Servlet

@ Java2 Platform 2

java.sun.com/products/hotspot



<http://java.sun.com/j2se/1.4.1/doc-files/sdk-jre.html>

@ Keywords ()

Keywords		
abstract	float	protected
boolean	for	public
break	future	rest
byte	generic	return
byvalue	goto	short
case	if	static
cast	implements	super
catch	import	switch
char	inner	synchronized
class	instanceof	this
const	int	throw
continue	interface	throws
default	long	transient
do	native	true
double	new	try
else	null	var
extends	operator	void
false	outer	volatile
final	package	while
finally	private	assert

Grayed words are reserved but not currently in use.

@ Identifiers

가

, - Pascal ex) PersonInSchool

, - Camel ex) moneyInMyPocket, goSchool()

가

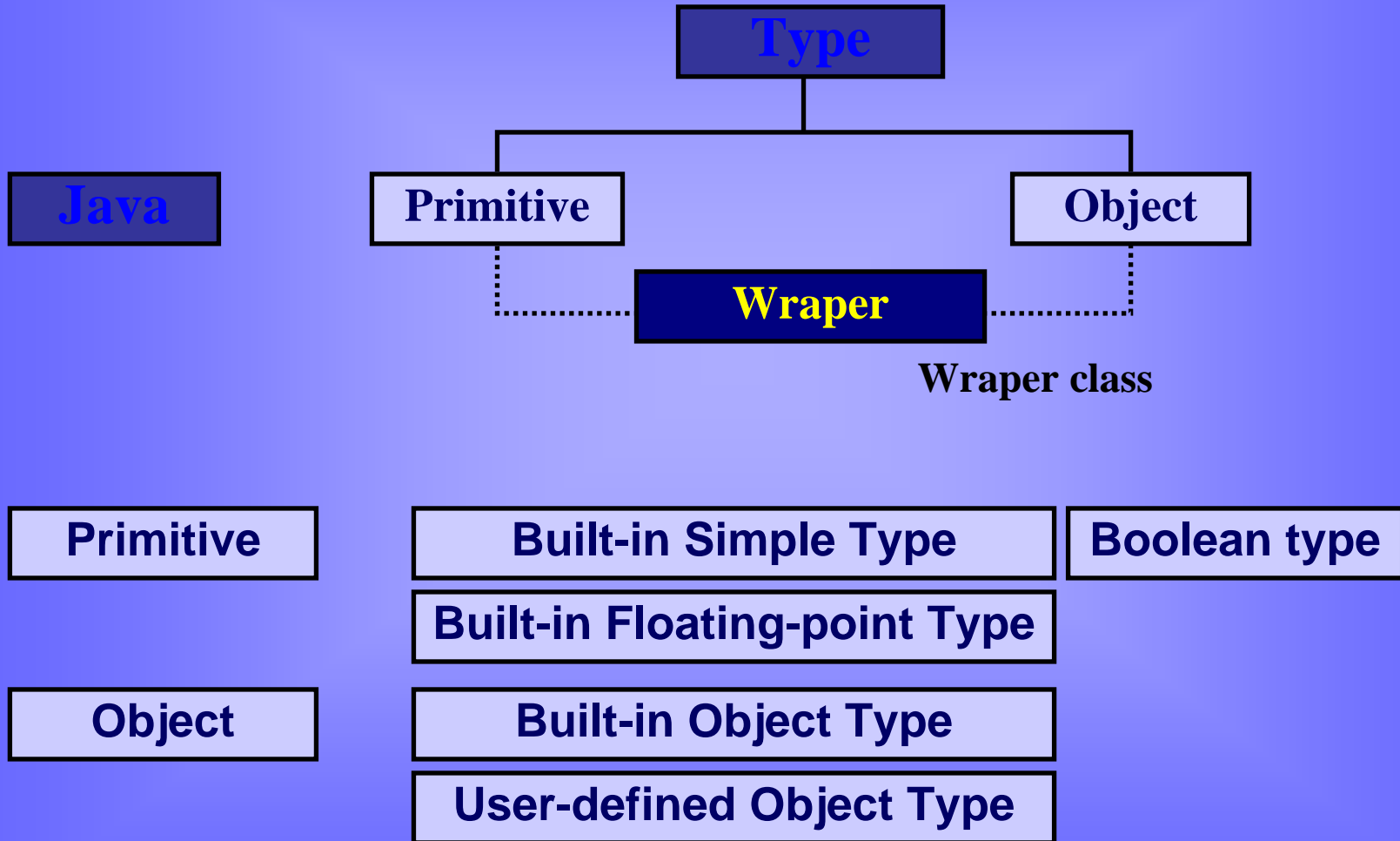
(,keyword 가)

\$(, _ (\$ nested class , _ c)

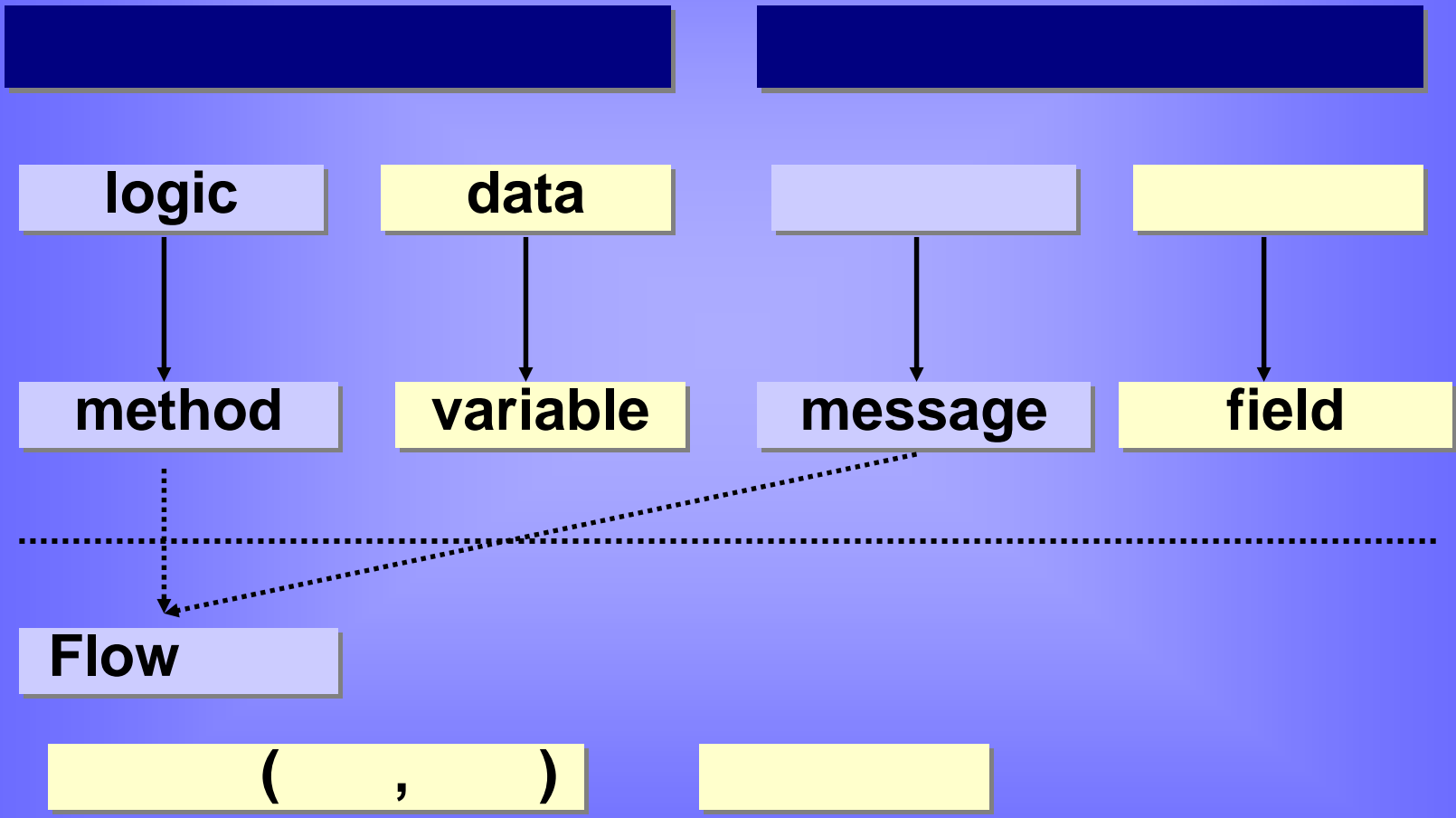
.(2 가 ,s2)

\$(, _ .

@ Type



@ vs.



@ OOP 3**OOO (Object Oriented Programming)****Encapsulation () :****Data (field)****Inheritance () :****member****Polymorphism () :****reference****argument****method**

@

downloading -

<http://java.sun.com/j2se/downloads.html>

installing

Configuration (path, classpath)

tool

editor –

MS

editor – BLUEJ (<http://www.bluej.org>)

@ Popular Java Development Tools

JBuilder - Borland

VisualCafe - Symantec

VisualAge - IBM

VisualJ++ - Microsoft

Java Development Kit(JDK) - Sun

AcroEdit – <http://www.acroedit.pe.kr/>

@ Let's try - Running Java Application

```
// HelloWorld.java
/* HelloWorld          */
class HelloWorld{
    // main          :          가
    public static void main(String args[ ]) {
        /** Hello World!          */
        System.out.println("Hello World!");
    }
}
```

Step 1. Compiling

```
> javac HelloWorld.java
```

Step 2. Run [Byte code]

```
> java HelloWorld
```

@ Let's try –

`System.out.print()` -
`System.out.println()` -
`System.in.read()` -

-1

```
public class InOutPractice{
    public static void main(String args[]) throws java.io.IOException{
        char ans;

        System.out.println("Do you want quit? (y/n)");
        while(true){
            ans = (char)System.in.read();
            if(ans == 'y') {
                System.out.println("You answered yes."); break;
            } else if(ans == 'n') {
                System.out.println("You answered no."); break;
            } else {
                System.out.println("You must use y or n.");
            }
        }
    }
}
```

Step 1. Compiling

> `javac InOutPractice.java`

Step 2. Run [Byte code]

> `java InOutPractice`

@ Let's try – Command Line Argument

`args[]` - 가

.

```
public class CommandLine{  
    public static void main(String args[]){  
        System.out.println("Hi! " + args[0]);  
    }  
}
```

Step 1. Compiling

> `javac CommandLine.java`

Step 2. Run [Byte code]

> `java CommandLine arg1 arg2 arg3 arg4`