

Lecture 06

Java – Why Java

Mostafa Elhosseini

Professor at College of Computer Science and
Engineering

Taibah University

<https://youtube.com/drmelhosseini>

| Why Java?

- The JAVA Programming language was designed to be
 - Object-oriented
 - Distributed
 - Simple
 - Multithreaded
 - Secure
 - Platform-independent

| Distributed



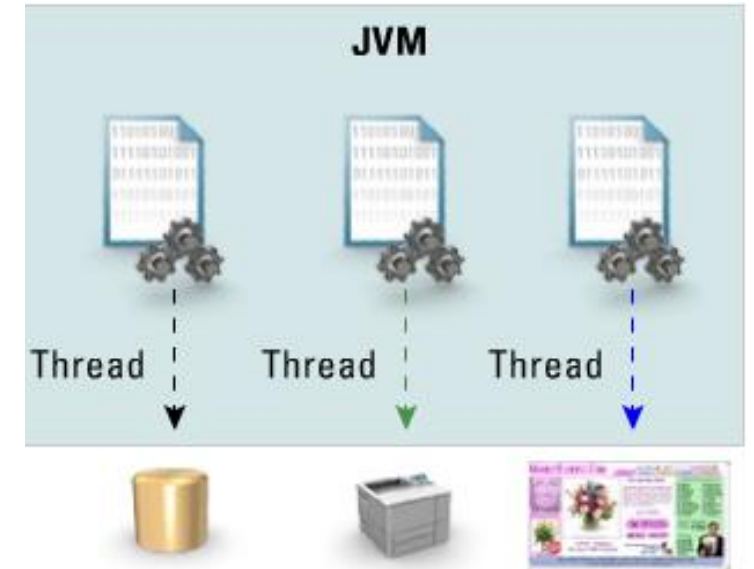
- Unlike programs in the past for all the pieces of the program had to be together before program execution,
- A JAVA program can actually have pieces scattered out of the place, even all may be on intranet and these done through things like
 - Remote Method Invocation (RMI)
 - Common Object Request Broker Architecture CORBA
 - Universal Resource Locator URL

| Simple

- Another big design consideration for the JAVA program is that Java is simple ,
- You see in a language like C++, you got lots of places you can suffer from troubles, you can access memory directly, allocate memory, deallocate memory, you still call multiple inheritance, you should go to these areas and minimize the risk involved
 - Reference instead of pointers
 - Garbage collector (no need to deallocate memory)
 - Single Inheritance
 - Boolean, either True, or false

| Multithreaded

- Computer users take it for granted that their systems can do more than one thing at a time
- One thread could access the database, another thread is printing something out to printer, and another thread obtain the user interface
- Can be used in Web server, database server
- The Java platform is designed from the ground up to support concurrent programming.



| Secure



- Every Java program is executed on what is called JAVA Virtual Machine (JVM)
- JVM interprets the Java code and execute them
- but before executing the code , it makes sure that the code does not have any thing that is unsafe in it, all the code is valid,

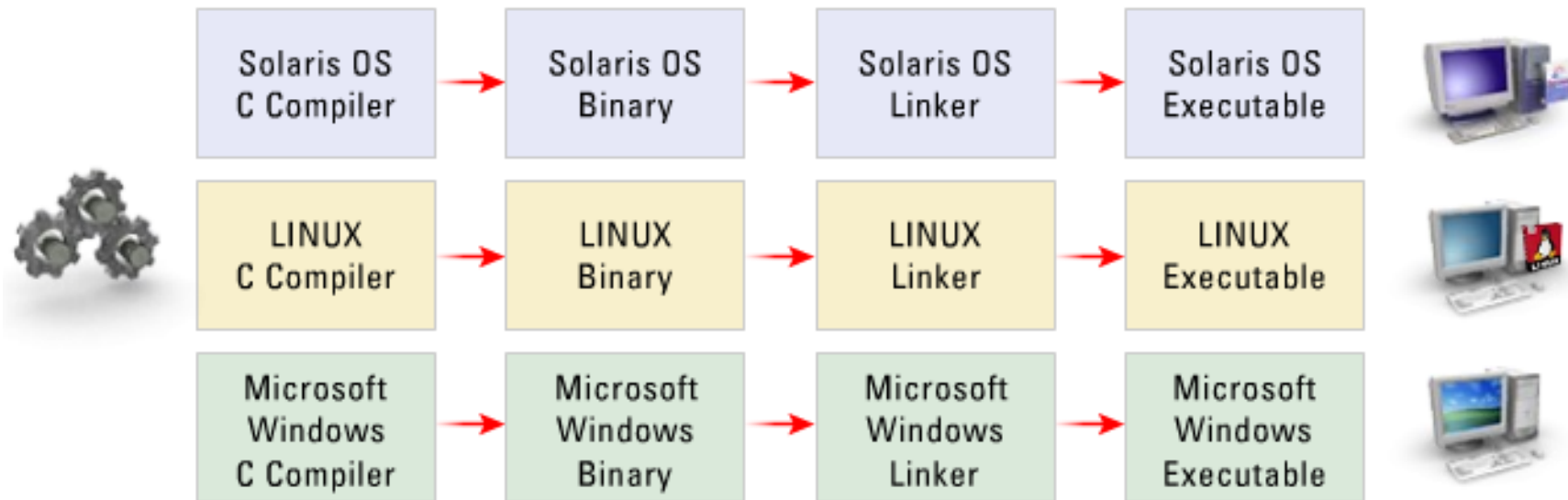
| Secure

- Java does not allocate direct pointers to memory. This makes it impossible to accidentally reference memory that belongs to other programs or the kernel
- Programs are prohibited from many activities, including:
- Reading or writing to the local disk
- Making a network connection to any host, except the host from which the applet came
- Creating a new process
- Loading a new dynamic library and directly calling a native method

| Platform-Independent

- Take a look back how program in languages in the past have been compiled and executed
- You see in the past, you take your source code, let's take a C language for instance, and run it through a compiler, and these compiler are specific to platform such as Solaris, LINUX, WINDOWS,
- The compiler will give a binary file that is platform dependent , it could be only executed on particular platform, ... etc

| Platform-Independent



| Platform-Independent



Java Compiler javac
(Solaris, Linux, or Windows)



Java Bytecode
(.class)



JVM Solaris JRE
java

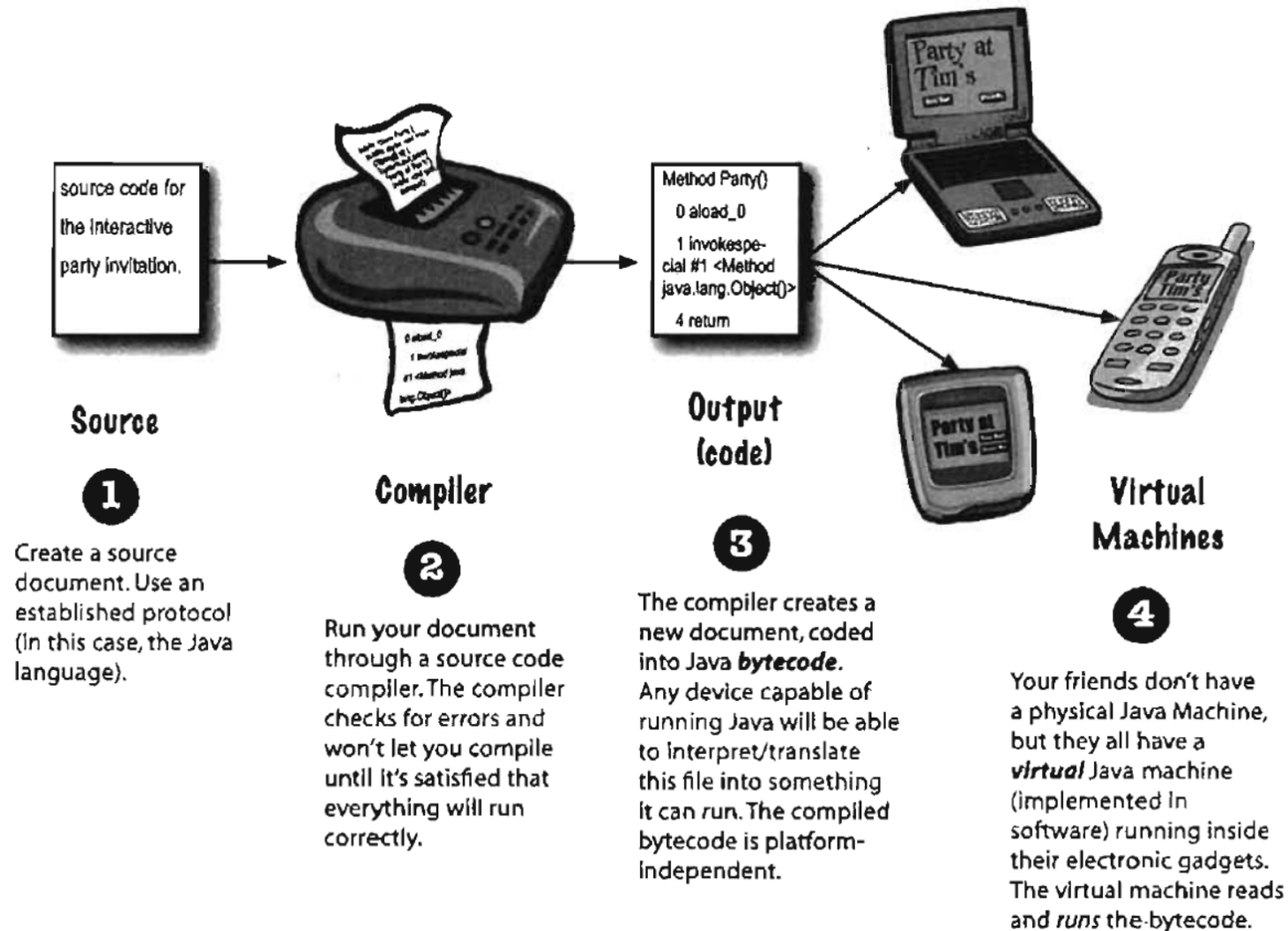


JVM Linux JRE
java



JVM Windows
java.exe JRE

| Platform-Independent



| Platform-Independent

- You see now, you are able to write a program once, compiled once and be able to executed on a varieties of platforms
- The source code is compiled and the compiler gives the byte code, and this byte code is very special, it is not for a particular platform, it is not for Solaris, it is not for LINUX, it is platform independent
- Byte code is interpreted by JVM, JVM is specific to given platform, one for Solaris, one for LINUX, one for Windows, and the byte code can be run on those different platforms and many others

JAVA Technology Product Groups



JAVA 2 Platform, J2SE

- Java 2 Platform, standard edition
JDK components
 - » Java Runtime Environment JRE
 - » Java virtual Machine (JVM) Java.exe
- Java Class Libraries
- Java Compiler – javac.exe
- Class Libraries documentation
(downloaded separately)
- Additional utilities
- Program examples



| References

- Java: How to Program, 9th Edition (Deitel) 9th Edition by Paul Deitel , Harvey M. Deitel
- Head First Java, 2nd Edition 2nd Edition by Kathy Sierra, Bert Bates
- Think Java: How to Think Like a Computer Scientist 1st Edition by Allen B. Downey, Chris Mayfield