



# **Explaining Java Technology**



# Key Concepts of the Java Programming Language

---

- Originated in 1991 as a research project to develop a language that would bridge the gap between consumer devices with different processors
  - Originally called Oak
- Initial concept failed but the authors recognized the language was perfect to develop Web multimedia components to enhance web pages
  - Applets became the initial use of the Oak language



# Java characteristics

---

- The Java programming language was designed to be
  - Object – oriented
  - Distributed
  - Simple
  - Multi-threaded
  - Secure
  - Platform-independent



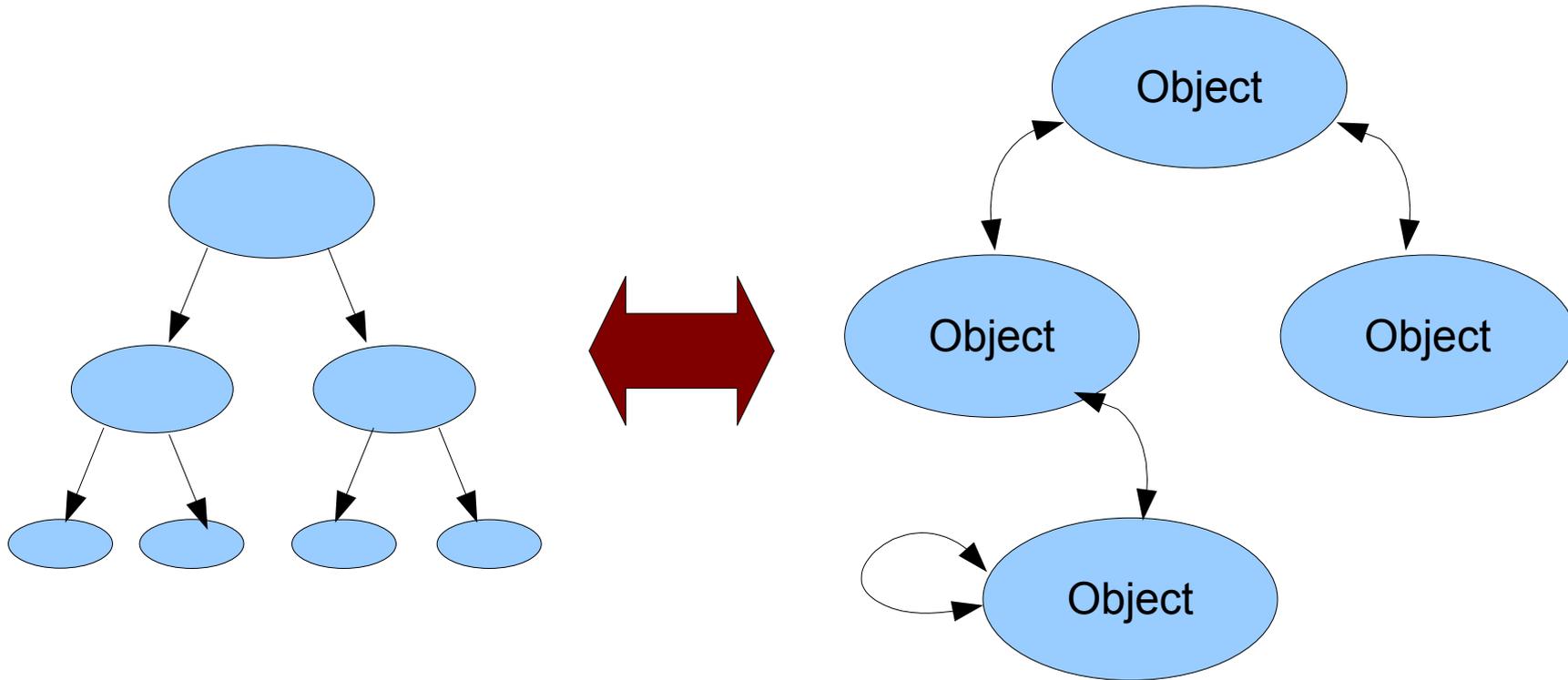
# Object-Oriented

---

- Java is an Object-Oriented Programming (OOP) language
  - The goal of the programmer is to create *objects*, that is pieces of code that can interact with other objects to solve a problem
  - OOP Started in 1967 with the Simula-67 language and has led to popular languages such as C++
  - OOP differs from *Procedural Programming* because procedural programming stresses the sequence of coding steps to solve a problem while OOP stresses the creation and interaction of objects.



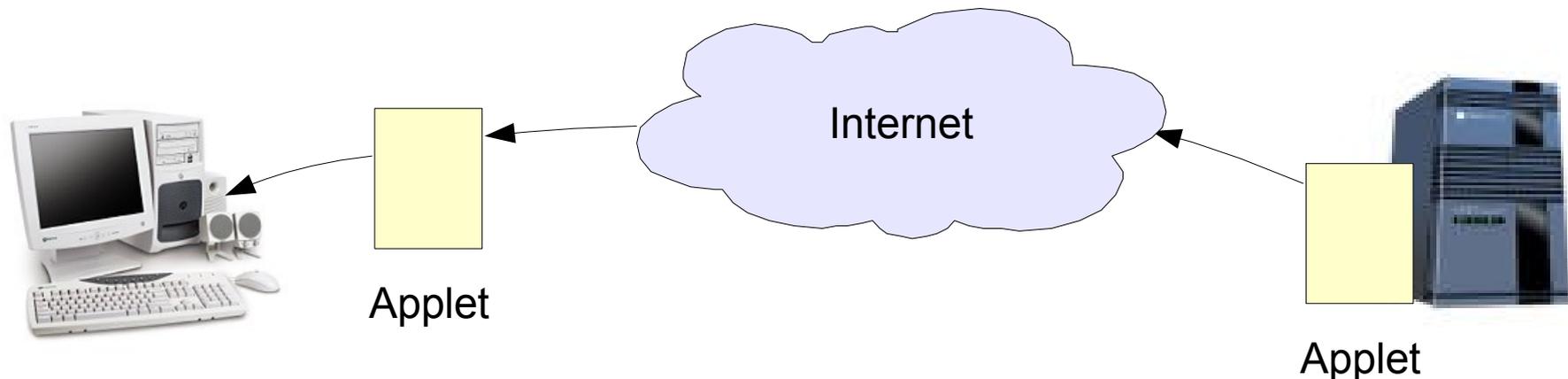
# Procedural vs OO





# Distributed

- Java is a distributed programming language because it provides support for distributed network technologies
  - Remote Method Invocation (RMI)
  - CORBA
- The dynamic class loading capabilities allow pieces of code to be downloaded over the Internet and executed on a personal computer.





# Simple

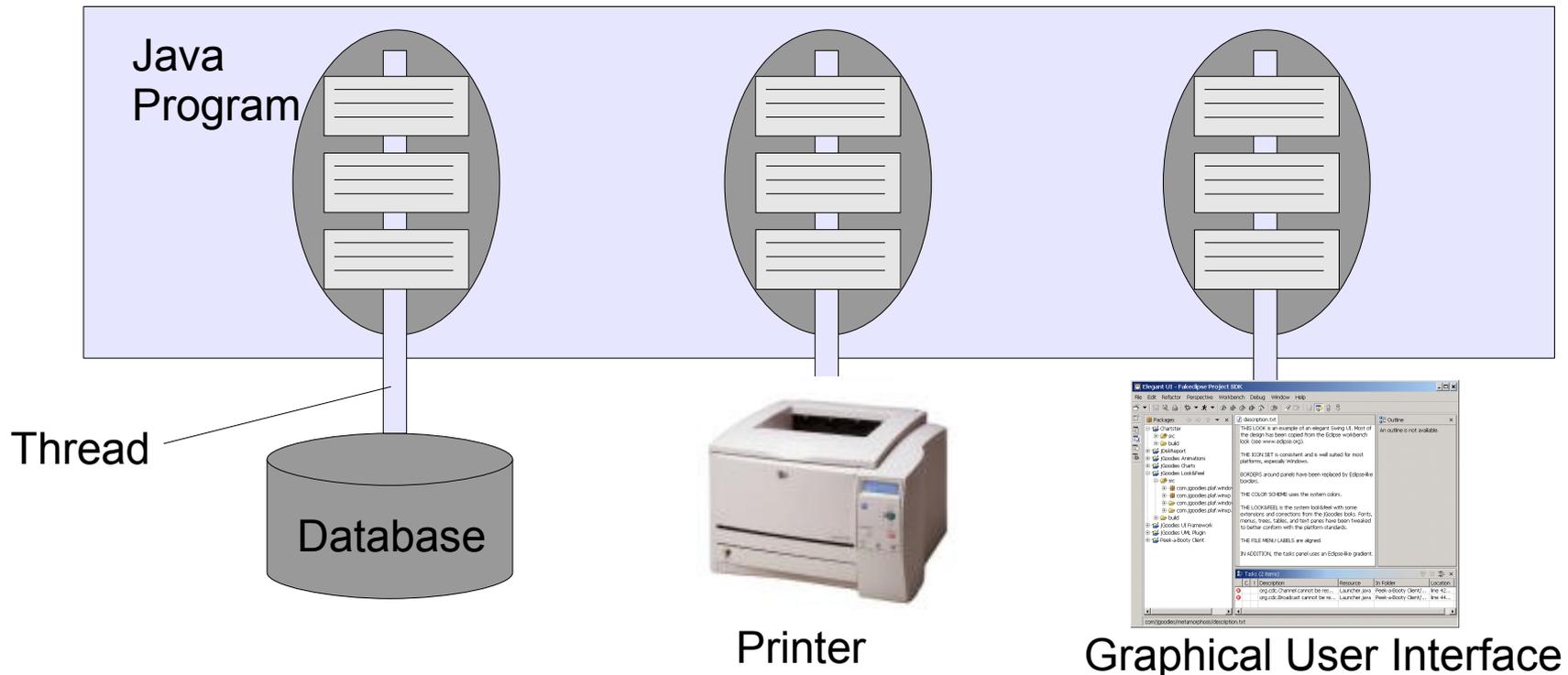
---

- The Java programming language is simple because its designers removed some of the complex or obscure programming constructs found in other popular programming languages
  - Pointers cannot be manipulated directly, objects can only be manipulated using *object references*
  - The *garbage collector* monitors and removes objects that are no longer being referred to
  - A boolean can only have a true or false value (not 1 or 0)



# Multithreaded

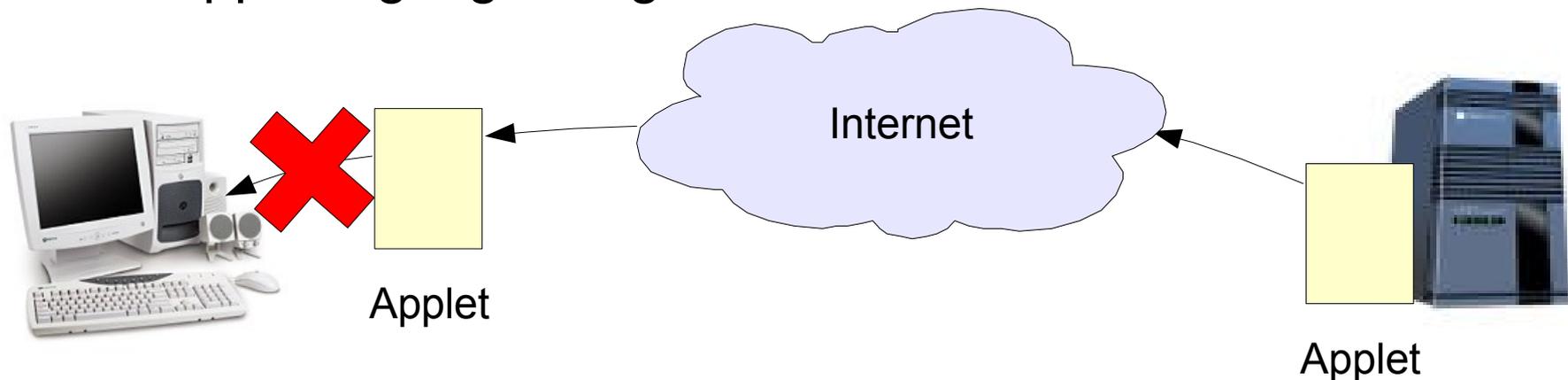
- The Java programming language supports multithreading, that is performing several tasks at the same time.
- Multithreading allows a program to be very efficient in its use of system resources





# Secure

- Java programs are secure because the environment in which the programs run uses security measures to protect programs from attacks.
  - Prohibiting the manipulation of memory using pointers
  - Prohibiting applets from reading or writing to the computer's hard disk
  - Verifying that the programs contain valid code
  - Supporting digital signatures





# Platform-Independent

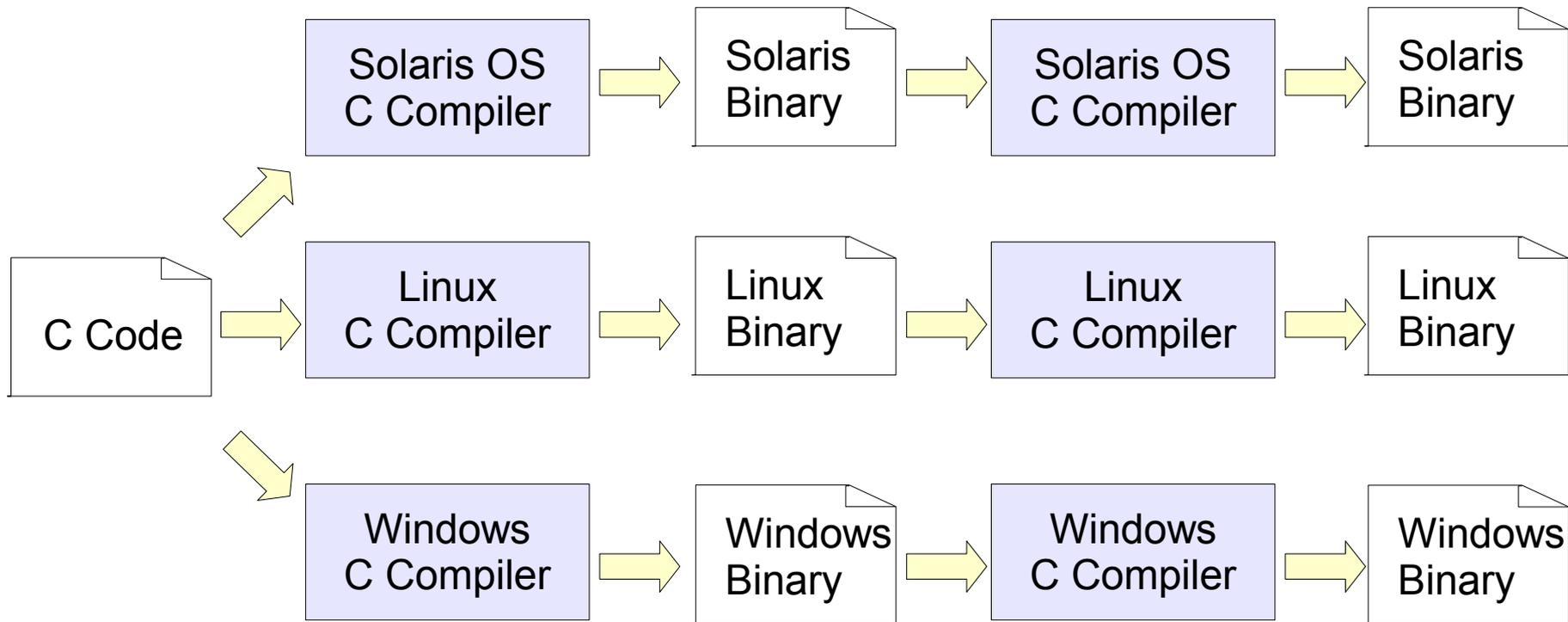
---

- Programs written in most languages require numerous modifications to run on more than one computing platform (CPU+OS).
  - Programming languages such as C and C++ require programmers to compile and link their programs, resulting in an executable program unique to a platform.
  - Unlike C and C++, the Java programming language is platform independent



# Platform-dependent programs

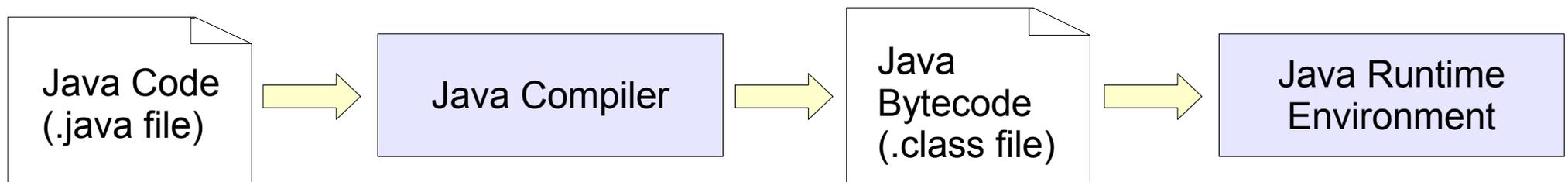
- Steps to create a platform-dependent executable





# Platform-independent programs

- The same Java program can run on several CPUs and operating systems with few or no changes
  - Java programs are also compiled using a Java compiler. The compiler produces platform-independent *bytecode* instead of CPU-specific code.
  - After the bytecode is created, it is interpreted (executed) by a bytecode interpreter called the *virtual machine*. The Virtual Machine is a platform specific program that understands platform-independent bytecode and can execute it.





# Java Runtime Environment

---

- A Java Virtual Machine (JVM) is required on every platform where the program will run.
  - The JVM is responsible for interpreting bytecode, loading java classes and executing the program.
- A Java program also needs a set of *standard class libraries*.
- Combined, the JVM software and class libraries are referred to as the Java Runtime Environment (JRE).
  - JRE's are available for many common platforms.



# Self-Check

- Match the terms with their definitions

## Definition

Composed of the Java Virtual Machine and class libraries

Manipulates objects instead of a pointer

Executes platform-independent bytecode

Removes unreferenced objects from memory

Executes within a web browser

Creates Java Bytecode

Executed by the Java Virtual Machine

## Term

Java Virtual Machine

Applet

Compiler

Reference

Bytecode

Java Runtime Environment

Garbage Collector



# Java Technology Product Groups

---

- Java technologies are included in three different groups of products, each designed to fulfill the needs of a particular target market.
  - *Java 2 Platform, Standard Edition (J2SE)* – Develops applets and desktop applications.
  - *Java 2 Platform, Enterprise Edition (J2EE)* – Creates large enterprise, server-side and client side distributed applications.
  - *Java 2 Platform, Micro Edition (J2ME)* – Creates applications for resource-constrained consumer devices.
- Only the class libraries change between editions



# Java 2 platform SDK

---

- The Java 2 Platform, Standard Edition Software Development Kit (SDK) includes the following:
  - Java runtime environment
    - A Java Virtual Machine for a specific platform
    - Java class libraries for the platform
  - A Java compiler
  - Java class library (API) documentation
  - Additional utilities, such as tools for creating Java archives (JAR files) and for debugging Java technology programs
  - Examples of Java technology programs