Mobile Application Languages and Framework– XML, Java, JME, Python and DotNet

Lesson 01

Java

App Program Development

- Using a set of statements, functions, methods, threads, objects, and classes
- Programming languages for apps are Java or J2ME, C#, Objective-C and Python

C and C++

- Programming languages used with Windows platform
- C and C++ with in-line-assembly, Visual Basic, and Visual C++

Java

- Class
- Thread
- Servlet
- Applet
- Package
- Compilation and Byte Codes

Class

- A construct in Java, which is used to define a common set of variables, fields, and methods and whose instances give the objects

Java and Java-based languages

- Possess some characteristic features
- Makes them the most used languages for mobile applications

Object oriented language Feature

- Each class– a logical group with an identity, state, and behaviour specifications
- A class is a named set of codes that has a number of members—fields, methods, etc.

Object oriented language Feature

- Objects can be created as instances of the class
- The operations done on the objects by passing the messages to them

Platform independence

- The program codes compiled in Java are independent of the CPU and OS used in a system because of the standard compilation into the byte codes
- Java codes complete mobility from one platform to another
- Java codes for mobile agents have weak mobility from one host to another

Robustness

- Java does not let the user assign pointers and write code for pointer manipulations
- All references to memory, freeing of the memory (garbage collection), de-allocation (de-referencing), and validation of object types and array indices done internally at the compile and run time

Standard APIs (application programming interfaces)

- Help a Java program to connect an application to a program, database, distributed object, or server developed on other platforms

Packages in Java

- Packages consisting of the classes and interfaces
- Packages help in fast development of the codes

Packages in Java

- java.lang for fundamental classes of Java

Packages in Java

- java.awt for Java foundation classes for creating GUIs

Packages

- java.swing for Java swing classes for creating lightweight GUIs
- Lightweight means not depending on the platform [system (OS) resources (for example, buttons)] unlike foundation classes which depend on the system resources

Packages in Java

- java.net is the package for network related classes for example, TCP/IP, UDP, HTTP, and FTP

Packages in Java

- java.security for classes in Java security framework. The examples are classes for RSA, DSA, and certificates
- java.sound for audio processing

Packages in Java

- java.io.Serializable– serializable interfaces

Serialization of Object

- Object state, as defined by the heap in memory, is written in memory as a serial byte stream
- Therefore, when the object migrates, it is received as serial byte stream and after de-serializing, the original object is restored into memory

JavaBean

- Has a *property* assigned to the fields in a class
- JavaBeans integrated into a *containe*r.
- Any change in property is listened to by a *listener* and an *action* method executes on change

JavaBean Example

- address property
- Any change in *address* has to be listened to by all related Beans in the container

JavaBean

- JavaBean– valuable for developing reusable software components
- The application, applets, or servlets integrate the JavaBeans
- java.beans package contains the classes for developing the JavaBeans

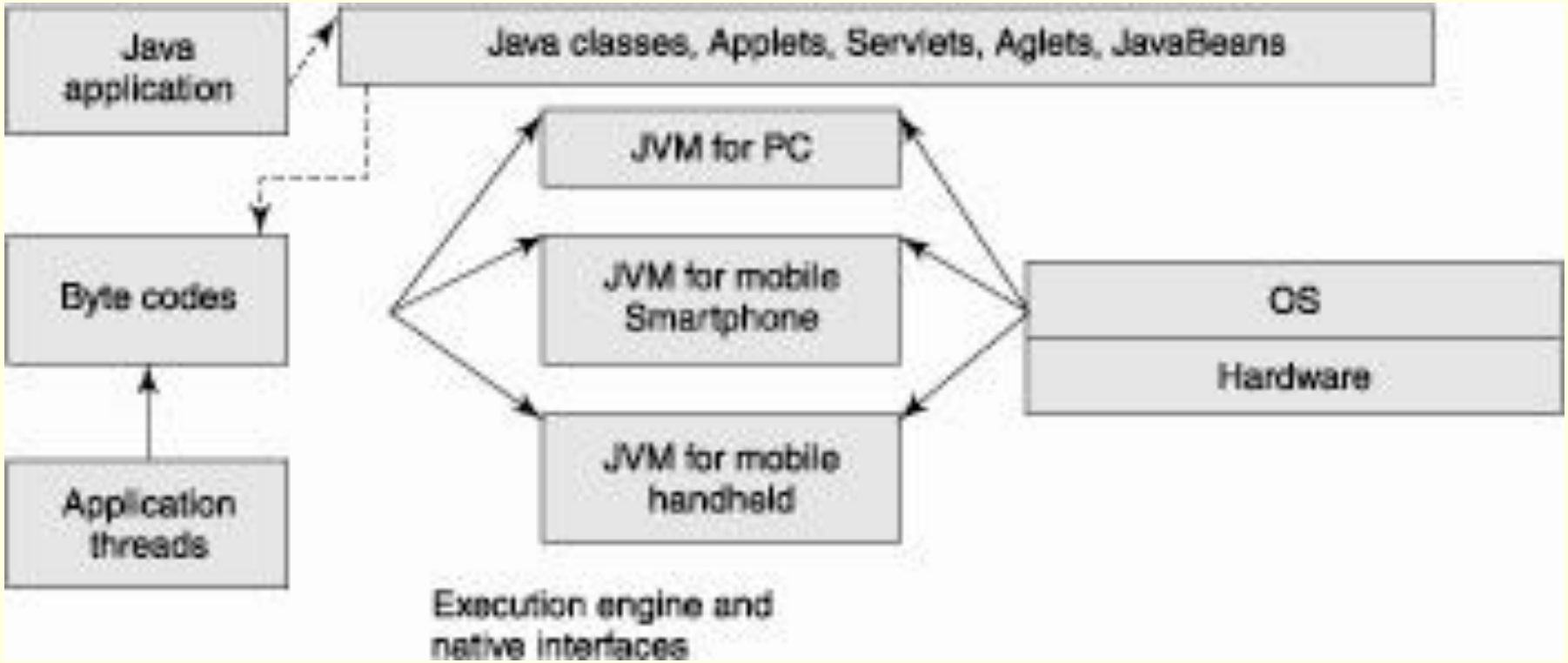
Java language

- Compiles into byte codes assuming a hypothetical CPU which possesses stack-organized architecture
- The codes run on a virtual machine
- Therefore, these can run on any operating system and hardware platform on which JVM is installed

JVM

- Provides the execution engine
- Native interfaces
- Runs the codes on the given OS and hardware

Java application, classes, and application threads compiled as platform-independent byte-codes



APPLICATION

- A program which runs on the user system
- Application threads– the programs of the application, each one of which is assigned a priority and runs concurrently

J2SE

- Used to program the *application*, threads, applets, servlets, and aglets

Servlets

- The programs running on a web server

Servlet Lifecycle

- Consists of the sequential steps—initiate, service, and destroy
- The init is analogous to main in a Java program
- Just as main ( ) executes first in Java program, init ( ) executes first in Servlets

Applets

- The programs running on a client
- Life Cycle of Applet consists of the sequential steps—initiate, start, stop, and destroy
- The init– analogous to main in a Java program
- Just as main ( ) executes first in Java program, init ( ) is executed first in Applet

Aglets

- The mobile agents which run on a host
- Can be retracted or cloned
- Can be used for messaging
- Supports weak mobility– can retain its state when it migrates from one host to another

Aglets Life Cycle

- Consists of the sequential steps—creation, dispatch, activation, deactivation, and disposition

An application

- Developed by programming a set of Components
- Database or server can be connected to the UI components using the APIs

Java applications

- Developed using open software Java 2 SDK from Sun. JBuilder, Microsoft Visual Studio J++, and IBM Visual Age
- IBM Visual Age provides a visual programming method to a programmer
- The Visual Age tool provides source code and compiled byte codes

Summary

- Java Byte code compilation
- Virtual machine
- Objected oriented
- Platform independent
- Robust
- Class, object and packages
- JavaBean

End of Lesson 01

Java