

■ Java

- Programming language for Internet environment (Sun Microsystems, 1995)
- Code execution on Java Virtual Machine
 - Multi platform (Unix, Windows, Macintosh, and others)
 - Small executables
- Java 2
 - Java Development Kit (JDK) versions 1.0 and 1.1 (Sun Microsystems)
 - JDK version 1.2 = Java 2 SDK (Software Development Kit) version 1.2
 - Many extensions and changes → old functions and styles will not be supported in future versions
 - Java 2 SDK version 1.3 → version 1.4.2
- Similar to C++, but simpler

■ Object-oriented programming (Java, C++, and others)

- Easy to re-use and maintain
 - Class – Instance
 - Modularization
 - Inheritance
- Program examples (“sample0.java,” “cylinder.java,” “fan_block.java,” “fan_block_smart.java”)
 - Bold blue: reserved words and characters

- Class – Instance

- Class (See files “cylinder.java” and “fan_block.java”)

- Description of general property and behavior for a category
 - ◆ Variable (Information)
 - ◆ Method (Operation)
 - Class definition
 - ◆ access class-name (variables, constructors, methods)
 - Method definition
 - ◆ access return-value-type method-name (arguments, operations)
 - Constructor
 - ◆ Special method to generate instances

- Instance (See file “sample0.java”)

- Specific example (with specific values) in a category

- Modularization

- Access (public, protected, private)

- Public: The variable or method is accessible from the outside
 - Private: The variable or method is not accessible from the outside

- Inheritance (See file “fan_block_smart.java”)

- Utilize predefined classes and add or redefine only those portions that are different
 - Efficient programming

- Add or override variables and methods
 - Compare “fan_block.java” and “fan_block_smart.java”

- Definition
 - Access class-name and extend super-class-name

- Miscellaneous

- Package
 - Group files based on directory hierarchy where the files are located
 - Avoid ambiguity (other classes (files) with the same name may exist)

- Import
 - Use other classes defined separately (in other files)

- How to make an object-oriented program complete your task
 - What information and operations does your task include?
 - How can the information and operations be modularized?
 - How can the information and operation modules be defined as class hierarchy?

- Preparation

- Download J2SE v 1.4.2 SDK (50MB) and J2SE v 1.4.2 Documentation (33MB) from Java Web page (e.g., “<http://java.sun.com/j2se/1.4.2/download.html>” for Unix and Windows)
- Install (If you are not an administrator, you need to ask one)

■ Sample program

- Download the following four files from the web page of this course
 - “sample0.java”
 - Make a “geometry” directory and place the following three files under it
 - “cylinder.java,” “fan_block.java,” “fan_block_smart.java”
- Compile
 - Open DOS window or Command prompt window
 - Move to the directory where the source file “sample0.java” is located
 - dir: list files
 - cd: change directory
 - Type: javac source-file-name (e.g., “javac sample0.java”)
 - Class files (e.g., sample0.class) are generated
- Run
 - Type: java class-file-name (e.g., “java sample0”)

■ J2SE 1.4.2 Documentation

- Tools such as javac and java → [Tool Documentation (docs)]
- Tutorials (website)
- Details of Java classes and methods → [Java 2 platform API Specification (docs)]