

COMP455- OBJECT ORIENTED PROGRAMMING

Using Java programming language

Dr. Constandinos X. Mavromoustakis



Personnel

□ Assistant Professor

Constandinos X. Mavromoustakis, BEng, MEng, MSc, PhD

- E-mail: mavromoustakis.c@unic.ac.cy
- E-mail is the best way to communicate with me
- During office hours,
 - feel free to stop by



Office hours

- **Student consultation hours:**

- *Monday 12:00-14:00*

- *Tuesday–Thursday 14:00-14:30*

- *Wednesday 12:00-13:30 && 18:00-18:30*

- **Contact me via e-mail**

- **Feel free to come in my office**



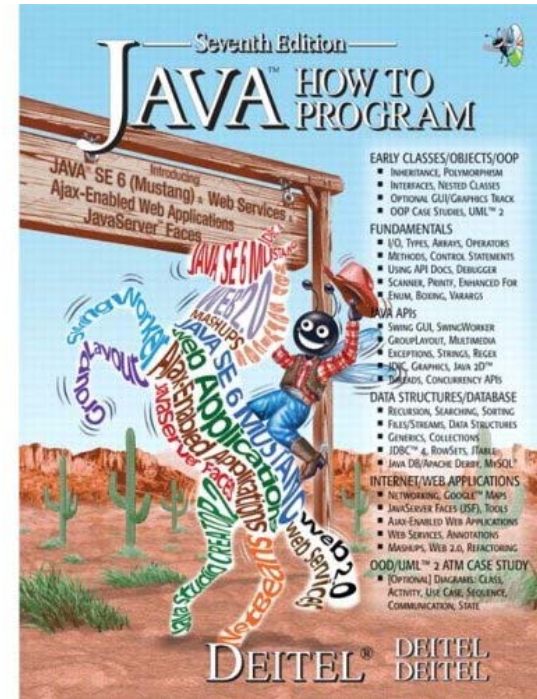
Course organization

- Please take a look @
 - Course outline
 - Web page:
 - *http://www.mavromoustakis.c.intercol.edu/COMP_455.html*

- 13 weeks of lecturing (NO. OF CREDITS:3)
- Quizzes...(+ some Homework)
- Programming assignment - obligatory

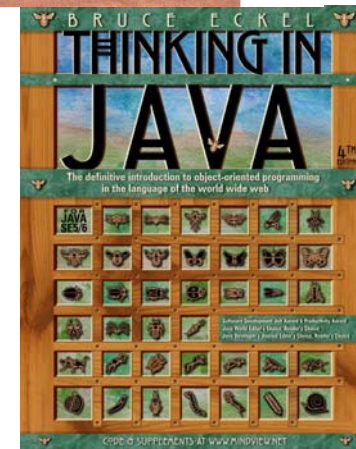
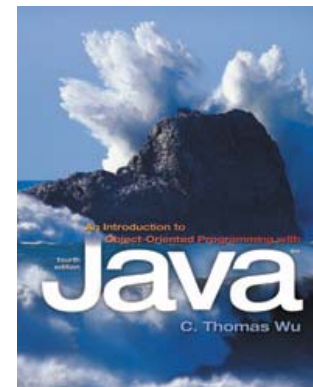
Reading

- Text book (main):
 - **Java How to Program (7th Edition), by Harvey M. Deitel, Paul J. Deitel**



Reference books:

- Thinking in Java, 4th edition , by Bruce Eckel
- An Introduction to Object-Oriented Programming with Java, 4/e , by C. Thomas Wu





Required skills


- The course assumes prior knowledge of programming.
 - *PREREQUISITE(S): COMP-255*
 - No need of knowledge of operating system principles.

- Algorithm concepts

- Also the basic features- what a computer system consists of (basic hardware parts) && Software parts (ex. compiler)

The aims of the course

Educational objectives:

- ❑ To learn the basic principles of the object-oriented programming with specific reference to the Java programming language
- ❑ To introduce and acquire the knowledge for simple object-oriented concepts and for more complex (private classes, objects, encapsulation, inheritance and polymorphism)
- ❑ To develop a way for efficient algorithmic thinking and problem solving using the object-oriented paradigm with the **UML** (Unified Modelling Language)
- ❑ To learn the Java graphical user interfaces (GUI) and the associated libraries of **SDK/Sun**  **JAVA**
- ❑ To describe, plan, and build simple applications using the concepts of object-oriented programming in the Java context. Student will be enforced to adopt object-oriented methods to a variety of problems, with emphasis on the Reverse Engineering (RE) paradigm

Java..

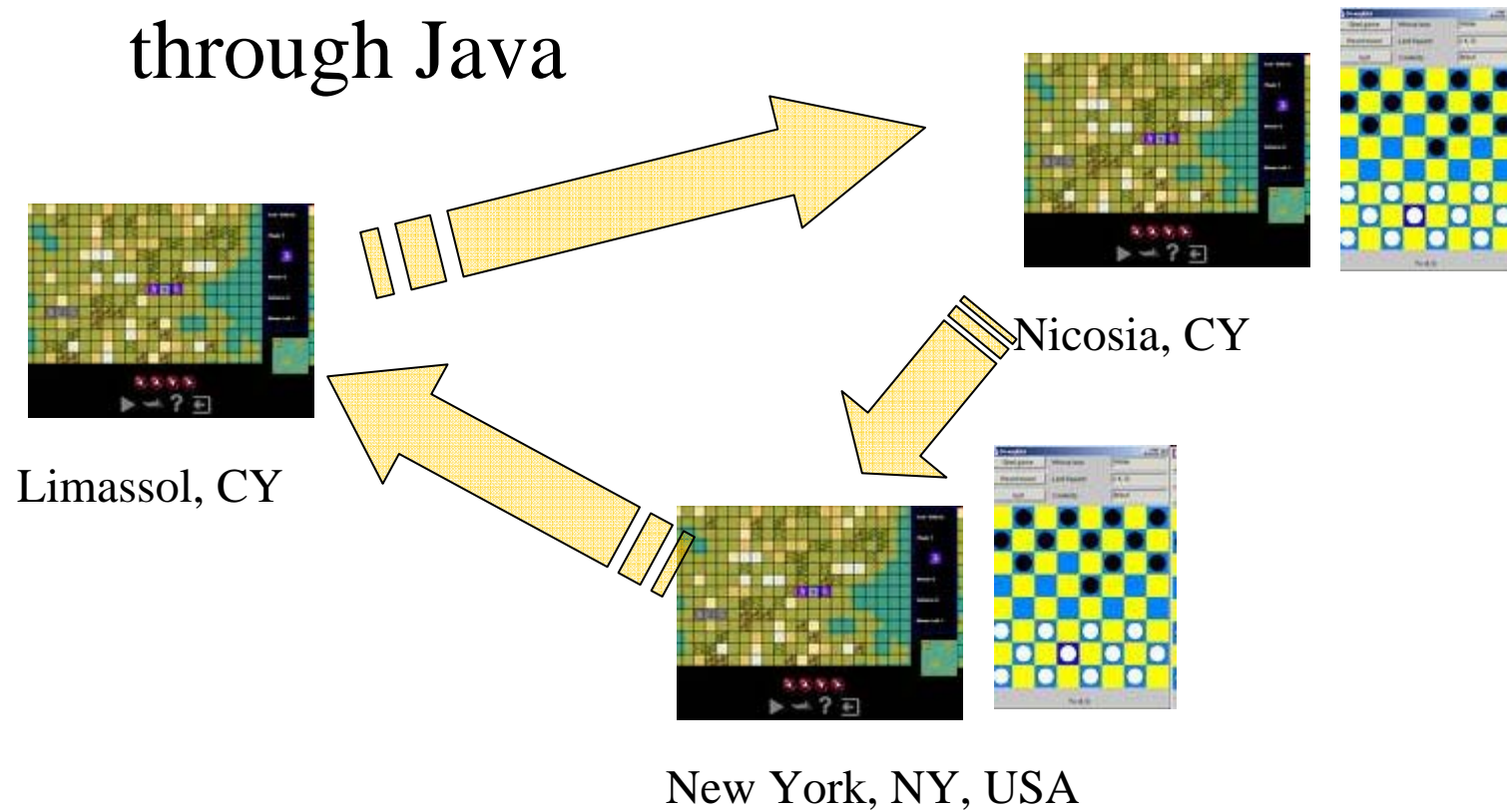


□ Almost everywhere

- Cell phones, web etc
- Vehicular java programs (GIS-Geographical Information Systems)
- Clock operations (UK, US, & some EU countries)
- TV texts and many other applications

Java...

- Remotely hosted machines communicate through Java





Programming...

- *Without a program, a computer has no purpose.*
- *Each machine is without “a spine” if there is no programming code in it.*
 - *Code == “a spine” for a specified purpose.*



Machine Languages, Assembly Languages, and High-level Languages

□ Three types of computer languages

1. Machine language

2. Assembly language

3. High-level languages

– Which one of the above is the most important?



Programming progression...

- Programming has progressed through:
 - machine code
 - assembly language
 - machine-independent programming languages
 - procedures & functions
 - objects

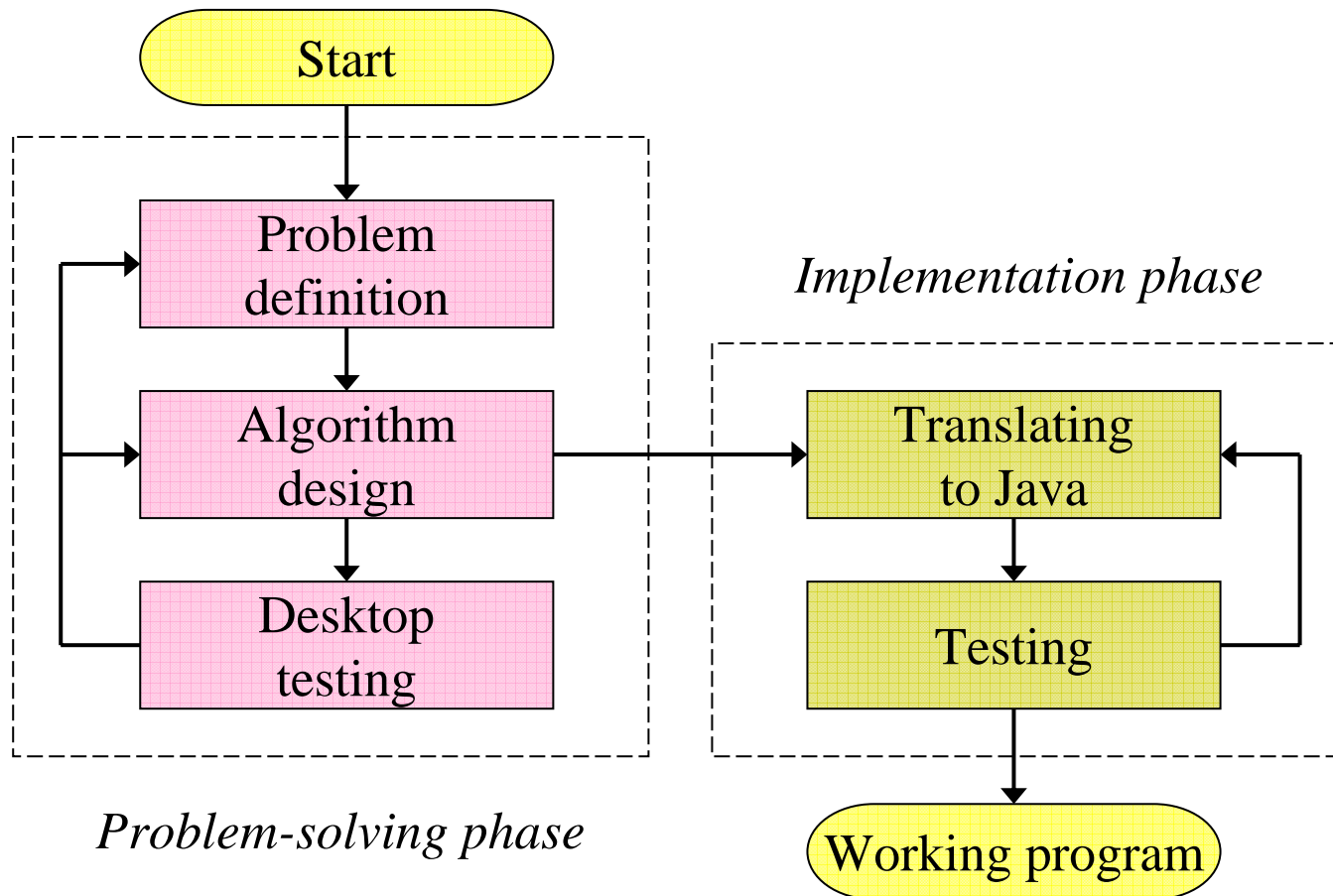


The Genesis of Java

- When the chronicle of computer languages is written, the following will be said: B led to C, C evolved into C++, and C++ set the stage for Java.

- Computer language innovation and development occurs for two fundamental reasons:
 - To adapt to changing environments and uses
 - To implement refinements and improvements in the art of programming

Software Engineering/ Simple problem solving hierarchy





What is Problem Solving?

- Problem solving
 - The process of taking the statement of a problem and developing a computer program that solves that problem

- A solution consists of:
 - Algorithms
 - Algorithm: a step-by-step specification of a function to solve a problem within a finite amount of time

 - Ways to store data



Textual or numeric

Specified actions (i.e. maths 30+12)

Program == data + operations

□ Programs manipulate data


- They simply:
 - Get some data
 - Perform operations on the data received
 - Provide results!



Programs with Object Oriented Design (OOD)

- Object Oriented
 - Object based modeling...
 - Real world objects that are translated into machine characters/language
 - Names are expressed as objects in machine language **BUT** in our minds have the meaning of a real world entity

Very Brief History of Java

- Started in 1991 by SUN Microsystems 
- Targeted at consumer electronics. Wanted reliable programming language
- Targeting stability and integrity
- Integrated into browsers
- Evolved into write once run anywhere, integrates into Netscape
- General purpose libraries released
 - Provide applications for consumer devices (cell phones, etc.)



What is java?

- ❑ Developed by Sun Microsystems (James Gosling)
- ❑ A general-purpose object-oriented language
- ❑ Based on C/C++
- ❑ Designed for easy Web/Internet applications
- ❑ Widespread acceptance



Why Java and Object Oriented Programming

- Inheritance
 - Cloning objects
 - Duplicate characteristics

- Less *codelines* with much more capabilities and much more **flexibility**



Features of Java

- Java syntax is similar to C/C++ but there are many differences too
- Java is strongly typed like C++
 - Type inconsistencies are detected at compile time
- Java is an Object-Oriented Language
 - Computation is achieved via *interacting objects* (more on this later)



Java Class Libraries

- Hierarchically structured...
- Classes
 - Include *methods* that perform tasks
 - Return information after task completion
 - Used to build Java programs
- Java contains class libraries
 - Known as Java APIs (Application Programming Interfaces)

Java Features



Java Features (1)

□ **Simple**

- fixes some clumsy features of C++
- no pointers
- automatic garbage collection
- rich pre-defined class library

□ **Object oriented**

- focus on the data (objects) and methods manipulating the data
- all functions are associated with objects
- almost all datatypes are objects (files, strings, etc.)
- potentially better code organization and reuse



Java Features (2)

□ **Interpreted**

- java compiler generate byte-codes, not native machine code
- the compiled byte-codes are platform-independent
- java bytecodes are translated on the fly to machine readable instructions in runtime (Java Virtual Machine)

□ **Portable**

- same application runs on all platforms
- the sizes of the primitive data types are always the same
- the libraries define portable interfaces



Java Features (3)

□ **Reliable**

- extensive compile-time and runtime error checking
- no pointers but real arrays. Memory corruptions or unauthorized memory accesses are impossible
- automatic garbage collection tracks objects usage over time

□ **Secure**

- usage in networked environments requires more security
- memory allocation model is a major defense
- access restrictions are forced (private, public)



Java Features (4)

□ **Multithreaded**

- multiple concurrent threads of executions can run simultaneously
- utilizes a sophisticated set of synchronization primitives (based on monitors and condition variables paradigm) to achieve this

□ **Dynamic**

- java is designed to adapt to evolving environment
- libraries can freely add new methods and instance variables without any effect on their clients
- interfaces promote flexibility and reusability in code by specifying a set of methods an object can perform, but leaves open how these methods should be implemented
- can check the class type in runtime



Java “Dis”advantages

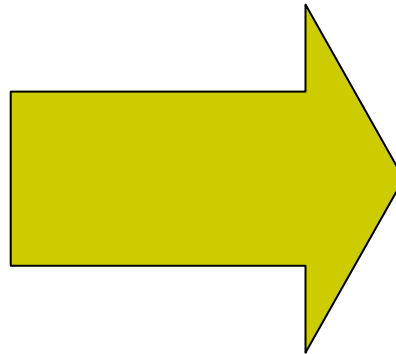
- **A bit slower than compiled language such as C**
 - an experiment in 2000 showed that Java was 3 or 4 times slower than C or C++

title of the article: “Comparing Java vs. C/C++ Efficiency Issues to Interpersonal Issues” (Lutz Prechelt)
 - adequate for all but the most time-intensive programs

Nowadays is almost the same in MOST cases!!!

The Java Buzzwords

- ❑ Simple
- ❑ Secure
- ❑ Portable
- ❑ Object-oriented
- ❑ Robust
- ❑ Multithreaded
- ❑ Architecture-neutral
- ❑ Interpreted
- ❑ High performance
- ❑ Distributed
- ❑ Dynamic



1. Java Multimedia Service
2. Java Web TV
3. Java E-Learning
4. Java Inter-Networking
- ...

Many many more..
Even we in this course,
can create one



Thinking About Objects

□ Objects

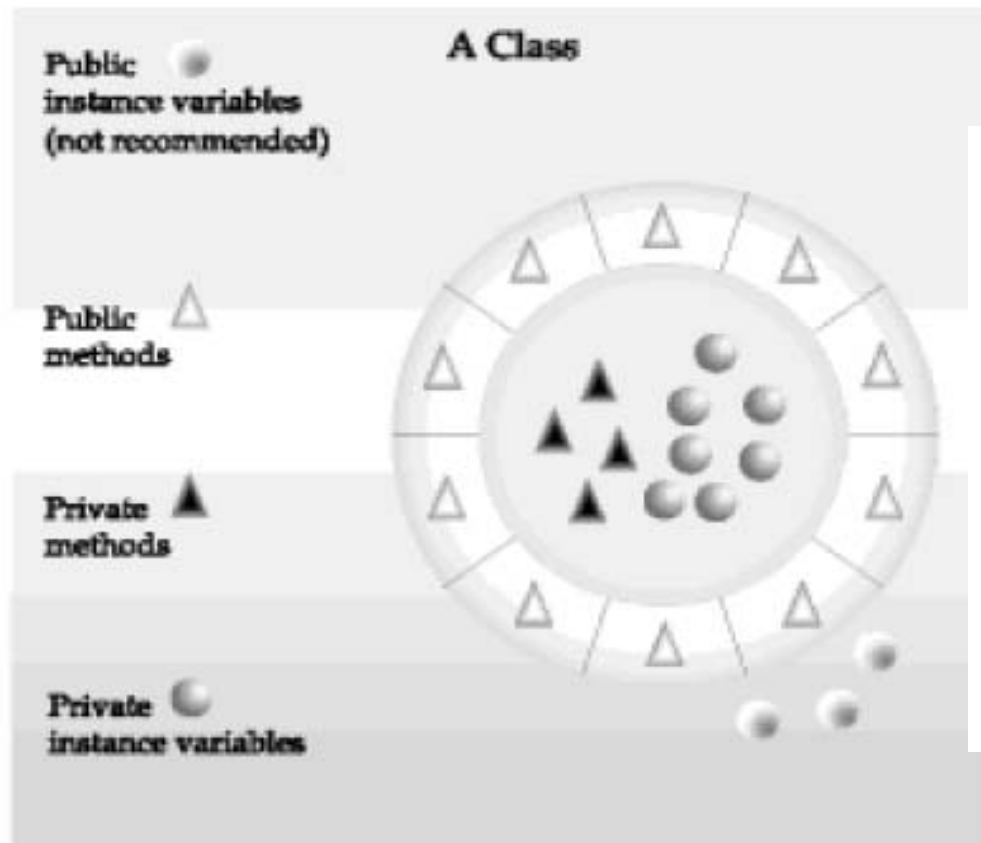
- Reusable software components that model real-world items
- Look all around you
 - People, animals, plants, cars, etc.
- Attributes
 - Size, shape, color, weight, etc.
- Behaviors
 - Babies cry, crawl, sleep, etc.



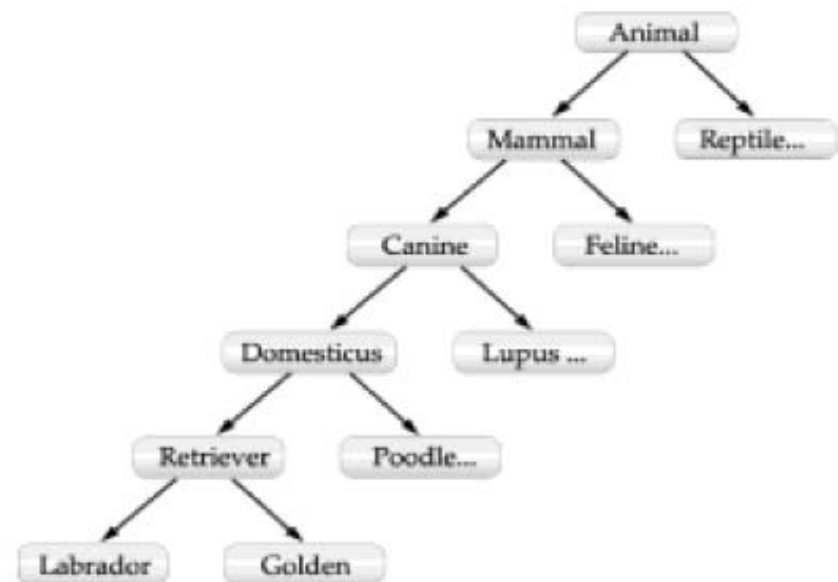
Thinking About Objects (cont.2)

- Object-oriented design (OOD)
 - Models real-world objects
 - Models communication among objects
 - *Encapsulates* attributes and operations (behaviors)
 - Information hiding
 - Communication through well-defined interfaces
- Object-oriented language
 - Programming in object oriented languages is called *object-oriented programming (OOP)*
 - Java

Java & Object Oriented Paradigm



class hierarchy





Basics of a Typical Java Environment

- Java programs normally undergo five phases
 - Edit
 - Programmer writes program (and stores program on disk)
 - Compile
 - Compiler creates *bytecodes* from program
 - Load
 - Class loader stores bytecodes in memory
 - Verify
 - Verifier ensures bytecodes do not violate security requirements
 - Execute
 - Interpreter translates bytecodes into machine language



Thinking About Objects (cont.3)

- Object-Oriented Analysis and Design (OOA/D)
 - Essential for large programs
 - Analyze program requirements, then develop solution
 - *UML*
 - Unified Modeling Language



Thinking About Objects (cont.4)

- UML
 - Graphical representation scheme
 - Enables developers to model object-oriented systems
 - Flexible and extendible

Programming...

- *Without a program, a computer has no purpose.*
- *Install Java VM as provided by D&D's CD, back cover of the book.*



- *http://www.murach.com/dloads/java/java_ch1.pdf*