CSE 2221 Software I: Software Components

and

CSE 2231 Software II: Software Development and Design

Restated Learning Outcomes

- Theme 1: ***software engineering concepts***
  - Be familiar with sound software engineering principles for component-based object-oriented software design

Software Engineering Concepts

- Component-based software engineering
  - System thinking
    - Mathematical modeling
    - Design-by-contract
    - Client vs. implementer view
  - Object-oriented software building blocks
    - Components and their relationships
  - Discipline
    - Single-point control over change
    - Adherence to conventions

Restated Learning Outcomes

- Theme 2: ***Java programming language***
  - – Be competent with Java programming

Java Programming Language

- Core syntax and features
  - Variables, types, values, operators, expressions, control flow (selection, iteration)
  - Reference vs. value types
  - Interfaces, classes, methods, objects
  - Inheritance, polymorphism
  - Generics, exceptions
- Libraries
  - Input/output, Java’s Swing for GUIs
  - Collections (e.g., List, Map, Queue, Set, …)

Restated Learning Outcomes

- Theme 3: ***industry-standard tools***
  - Be familiar with the use of industrial-strength software development tools

Industry-Standard Tools

- Eclipse
  - Industrial-strength open-source IDE
  - Many (free) plug-ins/extensions, including Checkstyle and FindBugs
- JUnit
  - Industry-standard library for unit-testing software components
- Javadoc
  - Industry-standard documentation utility for Java programs

Restated Learning Outcomes

- Theme 4: ***professional best practices***
  - Be familiar with Java programming "**best practices**"

Professional Best Practices

- Problem
  - Complex language mechanisms make it easy to produce code that is wrong, brittle, inextensible, and hard to maintain
- Solution
  - Discipline that helps (but does not guarantee) that developers write better code
- Examples
  - Naming conventions, coding conventions
  - Design-by-contract and programming-to-the-interface

Prerequisites

- Previous programming experience
  - Syntax, compilation, execution
  - Variables, types, expressions
  - Control flow (if, if-else, while, etc.)
  - Procedures/functions/methods
- Math maturity (introductory calculus)
- Ability/willingness to learn on your own
  - Goal: develop “life-long learning” capabilities

Resources

- Class meetings
  - Ask questions!
  - Answer questions!
- Instructor and grader
  - Make sure they know you by name
  - Visit during office hours or make appointment
  - Ask questions!
  - Answer questions!

Resources

- Course web site
  - http://cse.osu.edu/software/
  - All materials and links
- Class discussion group on Piazza
  - http://piazza.com/
  - A non-threatening forum for “anytime” Q&A
- Class website on Carmen
  - http://carmen.osu.edu/
  - Assignment submissions
  - Grades

Resources

- Online Java tutorials
  - http://docs.oracle.com/javase/tutorial/index.html
- Online OSU CSE components API
  - http://cse.osu.edu/software/common/doc/
- Online Java libraries API
  - http://docs.oracle.com/javase/8/docs/api/
- Many other Java resources available on the web!

Resources

- Many Java books available for free to OSU students via Safari Books Online

  http://proquest.safaribooksonline.com.proxy.lib.ohio-state.edu/

- Recommended books
  - C.S. Horstmann, *Big Java Late Objects*, John Wiley and Sons, 2012
    http://proquest.safaribooksonline.com.proxy.lib.ohio-state.edu/book/programming/java/9781118087886
  - J. Bloch, *Effective Java*, 2nd ed., Prentice Hall, 2008
    http://proquest.safaribooksonline.com.proxy.lib.ohio-state.edu/book/programming/java/9780137150021/