Course Organization

Object-Oriented Programming

SoftEng
http://softeng.polito.it

Teaching Staff

- Teacher
  - Silvano Rivoira
  - Dip. Automatica e Informatica
  - silvano.rivoira@polito.it
  - http://staff.polito.it/silvano.rivoira/
- Teaching Assistant
  - Leonardo Regano
  - leonardo.regano@polito.it
- Lab Assistants
  - Giuseppe Ministeri
  - Fabio Ricciardi

Topics

- Software Engineering
  - Software Life Cycle
  - Design
  - Test
  - Configuration management
  - Object-oriented paradigm
- Java programming language
  - Java syntax
  - Standard libraries

Objectives

- Understand how software development works
- Become familiar with the basic development support instruments
- Know the Java language
- Write and test simple Java programs
- Use the development support tools

Requirements

- Mandatory
  - Procedural programming (e.g. C)
- Recommended
  - Abstract data types
    - Lists, trees etc.
  - Algorithms
    - Sort, search, list insert etc.

Organization of the course

- Lectures (~50h)
  - Software Engineering (~15h)
  - Java (~35h)
- Classroom exercises (~20h)
  - Examples (~10h)
  - Assignments solutions (~10h)
- Lab assignments (~15h)
  - Two groups on alternating weeks
  - Three hours slots

Labs

- LAIBs
  - 1.5h with Teaching + Student Assistants
  - 1.5h with Student Assistant
- Assignments
  - Programs to be completed/modified
  - Similar process as in the final exam
- Assessed but not graded
  - *essential* for final exam:
  - the only way to learn programming is by… programming

Software

- Mandatory
  - Java 8
    - http://www.oracle.com/technetwork/java/javase/
  - Eclipse IDE (Java IDE)
    - http://www.eclipse.org/ide/
  - Subversive plug-in for Eclipse
    - https://www.eclipse.org/subversive/installation-instructions.php
- Useful
  - Astah – Community edition
    - http://astah.net/editions/community
  - Papyrus plug-in for Eclipse

Final Exam

- Part I: Software Development (~85%)
  - Step I: in the lab write the code
  - Step II: at home fix the code
- Part II: Theory (~15%)
  - Closed answer questions
- Rules
  - 2 hours
  - Books and notes are NOT allowed

Final exam – Development

- In the lab
  - Develop Java application, given
    - a textual specification of requirements
    - a skeleton code for the main functions
  - Submit initial version
- At home
  - Receive acceptance tests
  - Fix the app
  - Submit final version
    - Within a 3-7 days deadline

Final Exam – Assessment

- Functional correctness
  - Proportion of tests passed by the program version delivered in the lab
- Distance from correct version
  - Amount of changes between lab version and final version

Readings – Java

- Java Documenation
  - http://www.oracle.com/technetwork/java/javase/documentation/index.html
- Arnold, Gosling, Holmes. "The Java Programming Language – 4th edition", Addison-Wesley, 2006
- R. Urma, M. Fusco, A. Mycroft. "Java 8 in Action: Lambdas, streams, and functional-style programming." Manning, 2015.
- Eckel, "Thinking in Java", Prentice Hall, 4th Ed., 2006
  - www.mindview.com/Books

Readings – Sw Engineering

- Bruegge, Dutoit. ***Object-Oriented Software Engineering Using UML, Patterns, and Java***. Pearson, 2009
- ***ISO/IEC/IEEE Std 12207-2008 for Systems and Software Engineering - Software Life Cycle Processes***
  - http://ieeexplore.ieee.org/document/4475826/

Readings – Testing

- **ISO/IEC/IEEE, Std 29119-1 Software and systems engineering – Software testing** – Part 1: **Concepts and definitions, 2013.**
- ISTQB, Certified Tester Foundation Level Syllabus, 2001
  - http://www.istqb.org/downloads/send/2-foundation-level-documents/3-foundation-level-syllabus-2011.html4

Readings – Config Management

- **Collins-Sussman, Fitzpatrick, Pilato. Version Control with Subversion, 2001**
  - http://svnbook.red-bean.com
- IEEE Std 828-2012 *Standard for Configuration Management in Systems and Software Engineering*, 2012
- Semantic Versioning
  - http://semver.org

Readings – Design

- M.Fowler, K. Scott, *UML Distilled*, 3rd ed. Addison-Wesley, 2003.
- E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley, 1995.
- E.Freeman, E.Freeman, K.Sierra, B.Bates. *Head First Design Patterns*, O'Reilly, 2004