

# Recap of Week 1

Upon completing this lecture, you are expected to understand:

1. *Design by Contract* (*DbC*): Motivation & Terminology
2. Supporting *DbC* (Java vs. Eiffel):  
*Preconditions*, *Postconditions*, *Class Invariants*
3. *Runtime Assertion Checking* of Contracts

# Overview of Week 2

$$\begin{matrix} 4 & 4.3 & 2 \\ 3 & + & 3 & = & 6 \end{matrix}$$

#	DATE	LECTURES	IPAD NOTES	STUDY MATERIALS
		<p><b>▶ Playlist for Week 2</b> (<math>\approx 4.3</math> hours) <math>\rightarrow</math> 2 Hours will be deducted from <u>W3</u> + <u>W4</u> + <u>W5</u></p> <ul style="list-style-type: none"><li>Lecture 2 - Part 0: Introduction and Overview of Week 2</li><li>Lecture 2a - Part 1A: Modularity, Modular Design</li><li>Lecture 2a - Part 1B: Abstract Data Types (ADTs)</li><li>Lecture 2b - Part 1A: Copying Objects - Ref. vs. Shallow vs. Deep</li><li>Lecture 2b - Part 1B: Copying Objects - A Diagrammatic Example</li><li>Lecture 2b - Part 1C: 1st-Level Change - Reference vs. Shallow</li><li>Lecture 2b - Part 1D: Beyond 1st-Level Change - Shallow vs. Deep</li><li>Lecture 2b - Part 2A: Caching <b>old</b> expressions in Postconditions</li><li>Lecture 2b - Part 2B: Advanced Examples, <b>old</b> within Quantifications</li><li>Lecture 2b - Part 2C: Complete Postconditions - Principle</li><li>Lecture 2b - Part 2D: Complete Postconditions - Wrong Imp., Incomplete Contract</li><li>Lecture 2b - Part 2E: Complete Postconditions - Reference</li><li>Lecture 2b - Part 2F: Complete Postconditions - Shallow vs. Deep</li></ul>	<p>INTRO</p> <p>PRE</p> <p>POST</p>	<ul style="list-style-type: none"><li><b>Tutorials</b><ul style="list-style-type: none"><li><b>Introductory Tutorial</b> (continued from W1)</li><li><b>iPad Notes</b></li><li><b>Written Notes</b></li></ul></li><li><b>Slides</b><ul style="list-style-type: none"><li>Lecture 2a: Modularity, Abstract Data Types (ADTs)</li><li>Lecture 2a: Modularity, Abstract Data Types (ADTs) [4-up]</li><li>Lecture 2b: Copying Objects, Writing Complete Postconditions</li><li>Lecture 2b: Copying Objects, Writing Complete Postconditions [4-up]</li></ul></li><li><b>Questions?</b><ul style="list-style-type: none"><li><b>Post Your Questions in this Document.</b></li></ul></li><li><b>Source Code</b><ul style="list-style-type: none"><li>Copying of Arrays: Reference vs. Shallow vs. Deep</li><li>Writing Complete Postconditions: 5 Versions of {BANK}.deposit</li></ul></li></ul>
2	WED, Sep 16 to TUE, Sep 22			

Definitely move on to a **Virtual Machine** or **Remote Labs** if you are still struggling with installing **EStudio**.

## Learning Objectives of Week 2

- Modularity, Modular Design (maintainable, extensible, reusable)
- Abstract Data Types (ADTs) vs. Java Interface
- 3 Levels of Object Copying
- Use of the old Keyword in Postconditions
- Writing Complete Postconditions using  $\forall$  vs.  $\exists$

## Milestones of Week 2

- Complete Lab0 (based on intro. tutorials)
- Complete Quiz1 (based on lecture series W1)
- Start Lab1 (manual & starter tests) ; Scheduled Labs
- Start Lecture Series W2 *↳ precise documentation*
- Office Hours