Java By Abstraction: Chapter 7

Software Development

Some examples and/or figures were borrowed (with permission)
from slides prepared by Prof. H. Roumani

Development Process

- Design
- Implementation
- Testing
- Deployment

Design

- How the system will work
- Algorithm to generate the desired output
- Outline delegation of tasks
- Identify needed classes, methods, and attributes
- Determine how data will be exchanged amongst the various components

Implementation

- Involves coding…
  - Existing classes can be used/extended to meet requirements
  - New class created from scratch
- …and unit testing
  - Functionality of classes are tested individually to ensure adherence to specifications
  - (more details shortly)

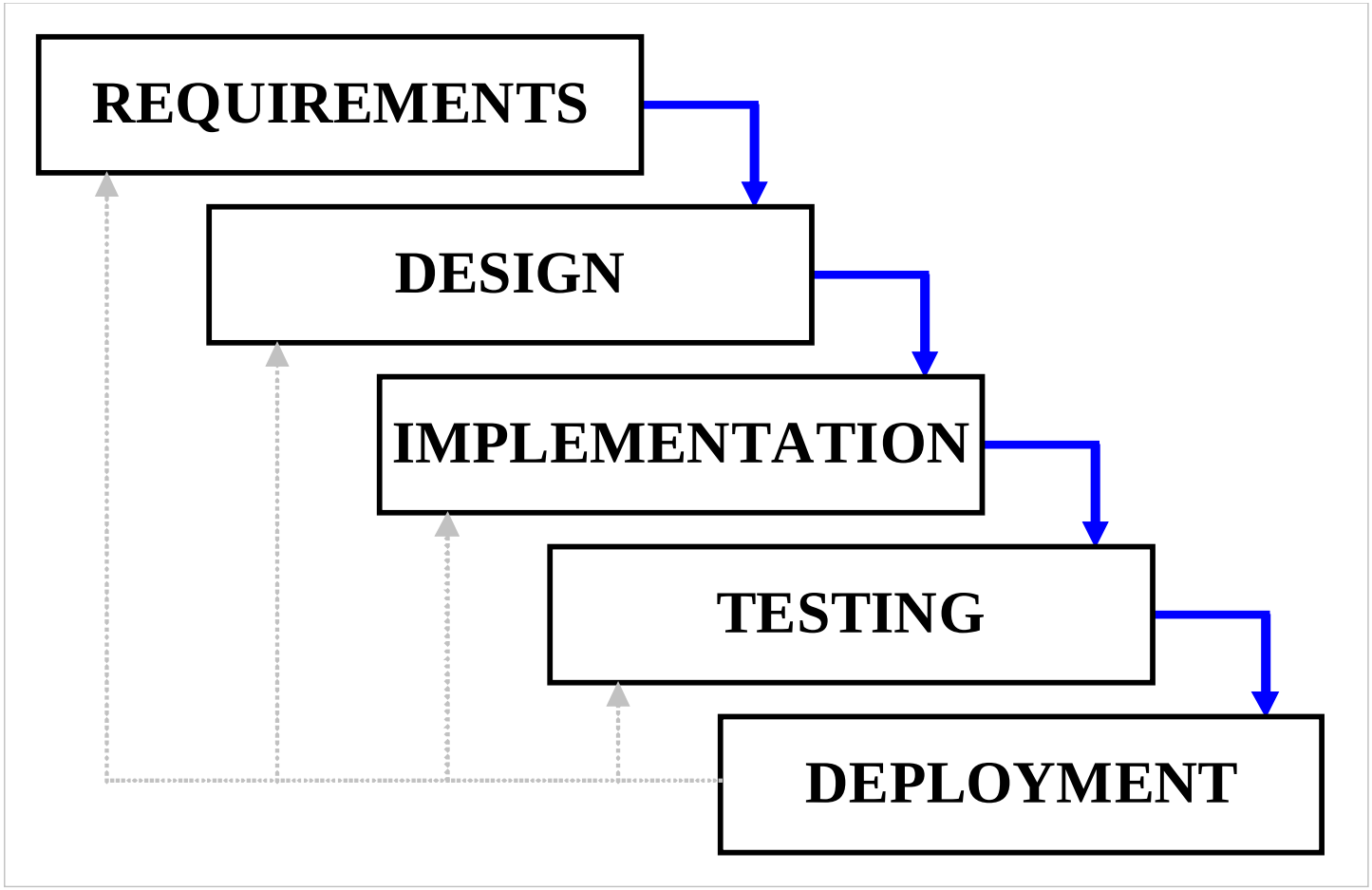
(Integration) Testing

- Evaluate entire system as a whole
- Ensure components work well together
- Ensure components exchange data correctly
  - Data formatting is especially important
  - Involves meeting specifications, not just “for looks”

Deployment

- Deployment
  - Package, deliver, and install system for customer
- Operation
  - Ensure functionality at the customer's location
  - Train customer's employees to operate system
- Maintenance
  - Develop and deploy updates, patches, and fixes
  - Perform upgrades

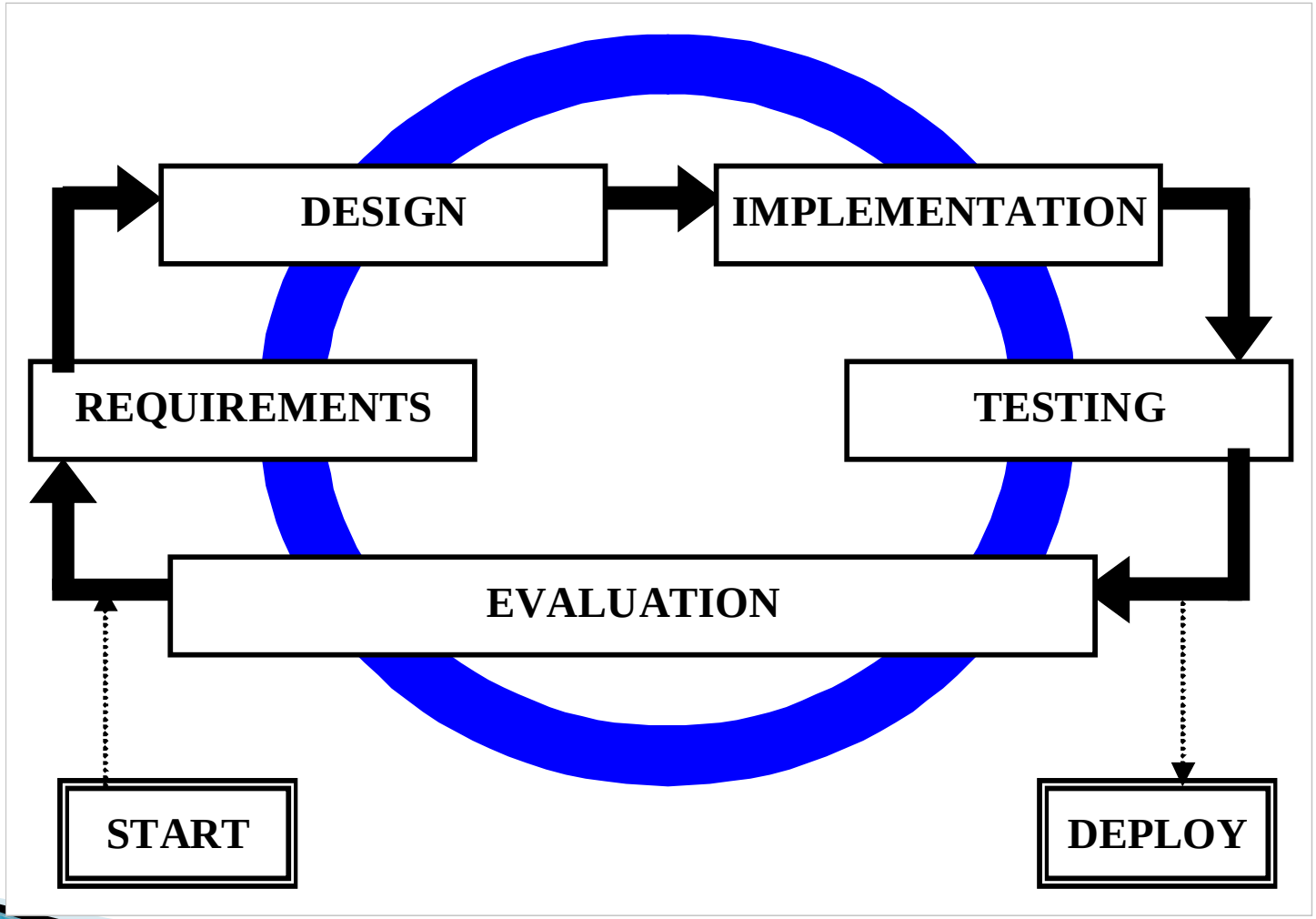
Waterfall Model



Shortcomings of the Waterfall Model

- Detection and handling of risks is delayed until the testing phase
- Risks include:
  - Interoperability problems amongst components
  - Requirement changes
  - Incorrect assumptions

Iterative Methodology



Iterative Methodology Models

- Agile software development
- Extreme Programming (XP)
- (IBM) Rational Unified Process (RUP)
- SCRUM development

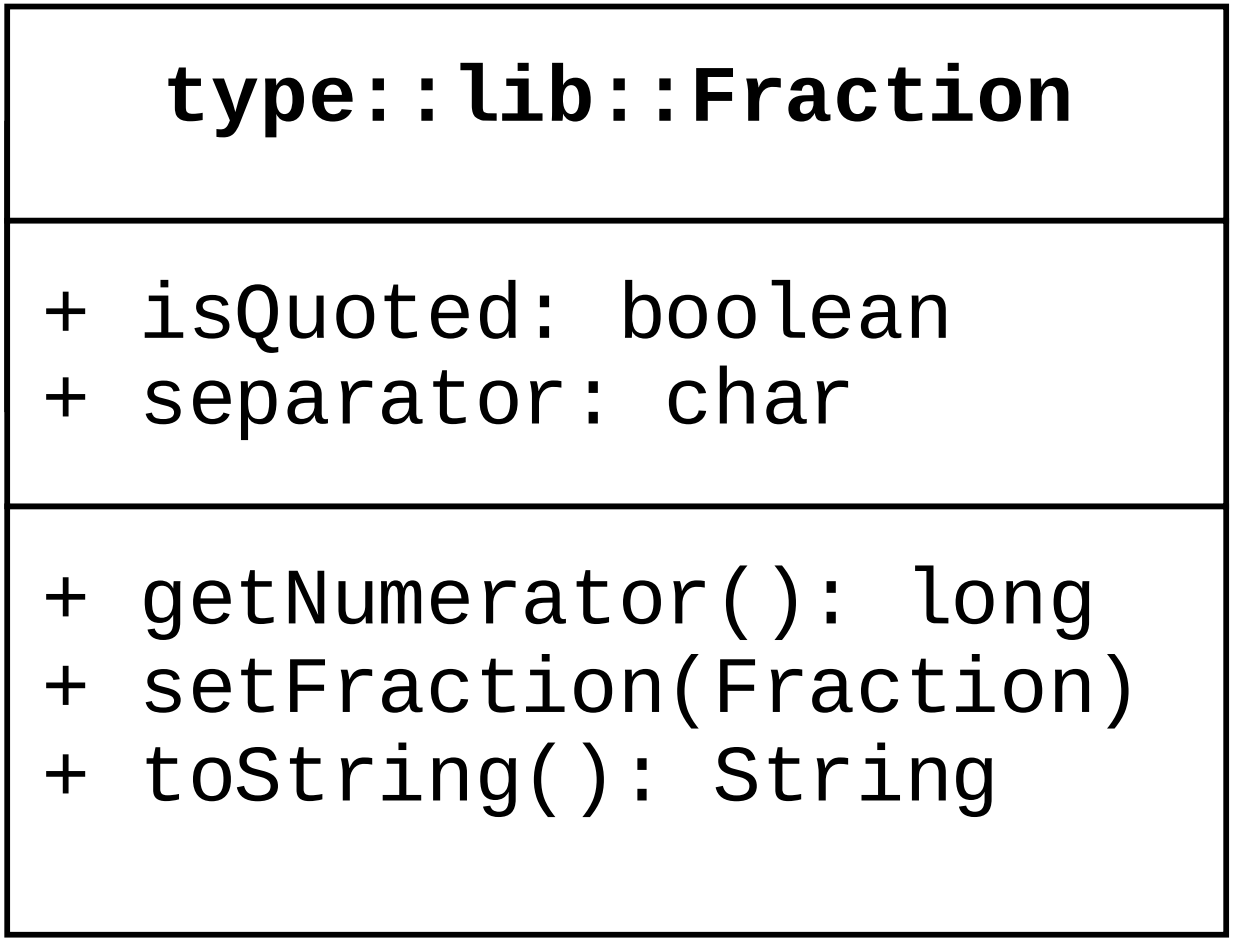
Design Techniques

Some examples and/or figures were borrowed (with permission)
from slides prepared by Prof. H. Roumani

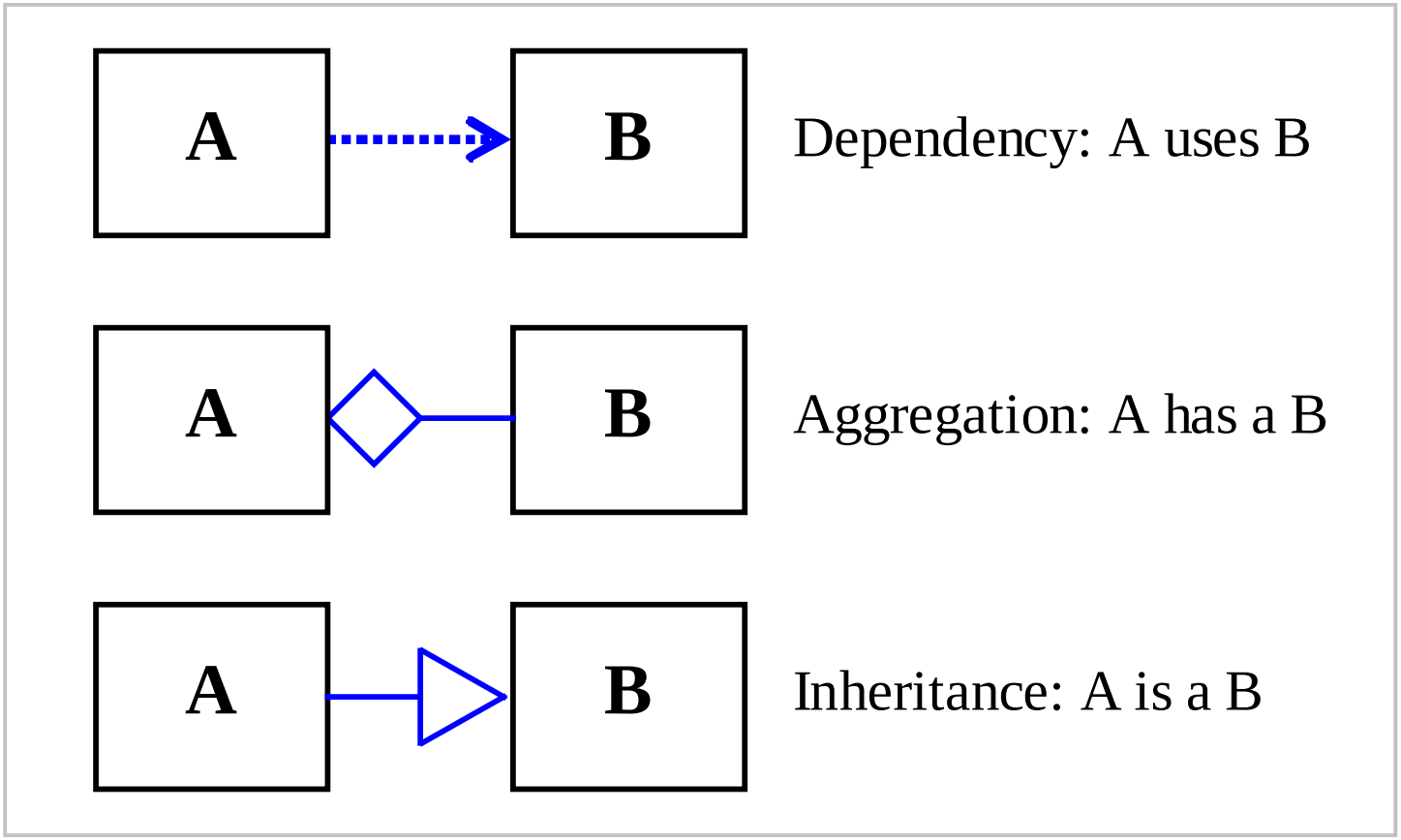
Unified Modelling Language (UML)

- Visual language used to describe characteristics and interactions of software components
- Formal language with rules but also flexible
- UML tools convert UML diagrams ↔ code

UML Diagrams



UML Relationships



Software Testing

Some examples and/or figures were borrowed (with permission)
from slides prepared by Prof. H. Roumani

Formal Proofs

- Written proofs using:
  - Discrete mathematics
    - Axioms
    - Theorems
- Covered in MATH1019 and MATH1090

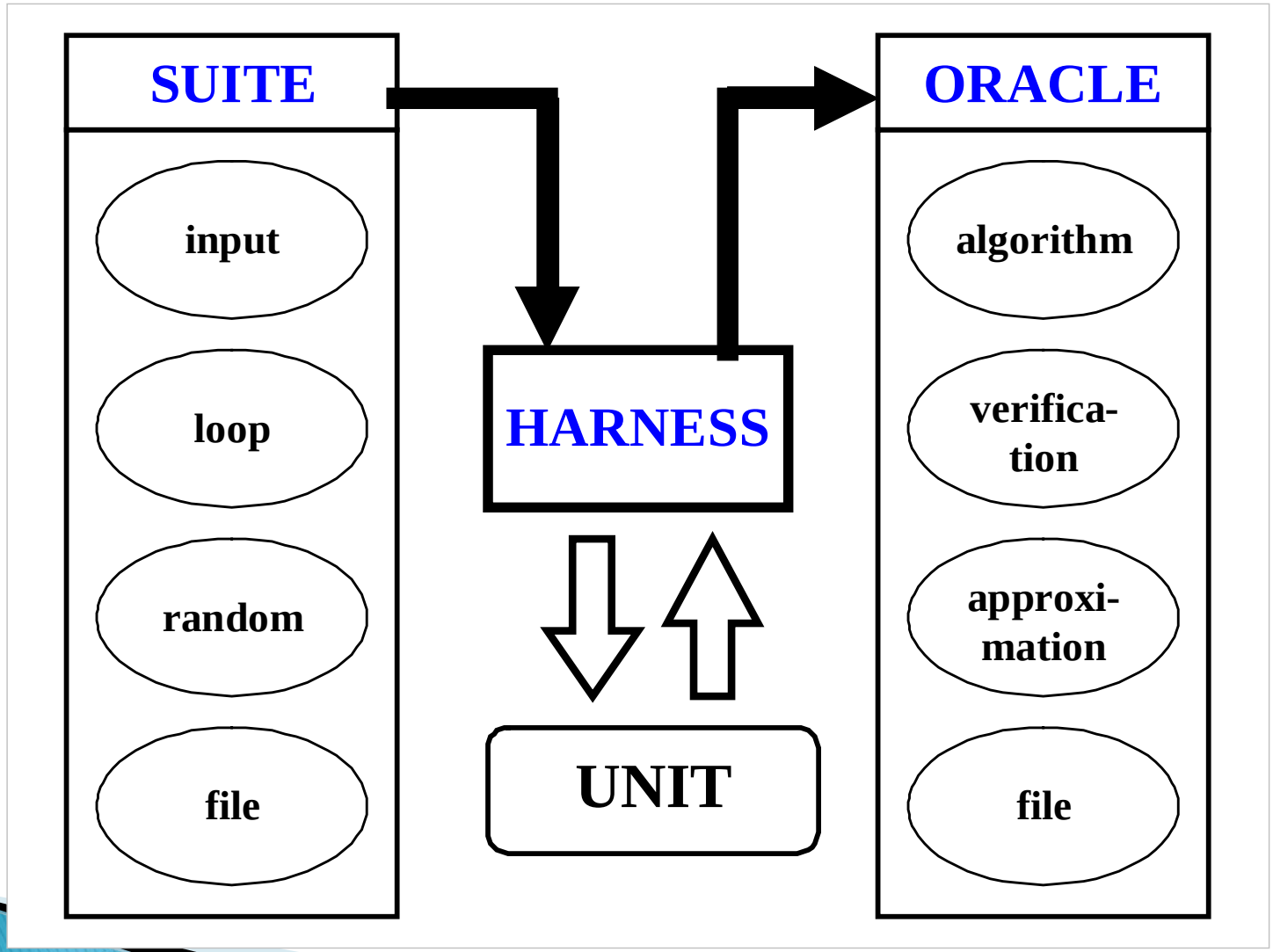
Test Vector

- Collection of test cases
- Test cases should include:
  - Values within range
  - Values outside range
  - Boundary cases
- Each case should hold meaning and test a specific aspect of the component
- Cover as many execution paths as possible
- Employ regression testing

Test Harness

- Program to automate the testing of a component
- Takes unit test input
- Compares component output to oracle's output
- Oracle:
  - Separate mechanism, component, or algorithm
  - Provides the "correct answer"
- Can you give an example of a test harness?

Test Harness



Debugging

- Determine and fix source of error
- Techniques:
  - Examine code (tedious, error-prone)
  - Print statements to output intermediate steps/values
  - Examine error messages for source details
  - Use debuggers