COMP2511

Object-Oriented Design and Programming

Programming by Contract

Wayne Wobcke

w.wobcke@unsw.edu.au

Today’s Lecture

- Java Type System
- Exceptions
- Defensive Programming
- Design by Contract
- Test Driven Development

Java Type System

- Everything is an Object . . .
  - Except int, float, double, etc., and null
- Assignment b = a is valid if actual type of a is a subtype of declared type of b
- Parameters are references and are called by value

Exceptions

- try – throw – catch – finally
- Hierarchy of exception classes
- Exceptions vs errors
- Checked vs unchecked

Defensive Programming

- What error checking would you have in BankAccount in first year?

Programming by Contract

- Pre- and postconditions
- Class invariant
- Don’t need to check preconditions in the called method (maybe in caller)
- Exceptions can be part of contract

Pre- and Postconditions

- Conditional contract
- **If** the precondition holds before a method call, **then** the postcondition holds after the method terminates
- Nothing guaranteed otherwise

Class Invariant

- Condition on object state
- Holds after constructor, and before and after each method call − but may not hold during method execution
- Reasoning: If precondition and invariant hold before method call, postcondition and invariant hold afterwards

Covariance

- If $s \leq t$ then $f(s) \leq f(t)$
  - $s \leq t$ means s is a subtype of t
  - if $f(s)$, $f(t)$ are propositions A, B, $A \leq B$ means A *logically implies* B
- f is postcondition of a method, the result type of a method, or the class invariant

Contravariance

- If $s \leq t$ then $f(s) \geq f(t)$
  - $s \leq t$ means s is a subtype of t
  - if $f(s)$, $f(t)$ are propositions A, B, $A \geq B$ means B *logically implies* A
- f is precondition of a method or the argument type of a method
- Liskov Substitution Principle

Test Driven Development

- **First** define a set of tests for all methods, **then** develop code to pass each test
- Is this a good idea?
- TDD does not just mean “I tested my code”

Next Week

- Object-Oriented Design Process
- Assignment 1