Design

Abstract Data types (ADTs)
Cohesion Principle
Single Choice Principle
Open-Closed Principle
Design Document
Justified Design Decisions

Design by Contract (DbC):
Class Invariant, Pre-/Post-condition
Information Hiding Principle
Eiffel Testing Framework (ETF)
Abstraction (via Mathematical Models)
Regression Testing
Acceptance Testing
Void Safety
Generics
Multiple Inheritance
Sub-Contracting
Architectural Design Diagrams

Eiffel

Syntax: Implementation vs. Specification
agent expression, **across** constructs
expanded types, **export** status
Runtime Contract Checking
Debugger

Architecture: Client-Supplier Relation
Architecture: Inheritance Relation
Program to Interface,
Not to Implementation
Modularity: Classes
Design Patterns
(Iterator, Singleton, State, Template,
Composite, Visitor, Strategy,
Observer, Event-Driven Design)
Anti-Patterns

Specification: ***Predicates***
Contracts of Loops: Invariant & Variant
Program Correctness
Weakest Precondition (**WP**)
Hoare Triples
Specification: Higher-Order Functions

Code Reuse via Inheritance
Substitutibility
Polymorphism (esp. ***Polymorphic Collections***)
Type Casting
Static Typing, Dynamic Binding
Unit Testing

OOP

Axioms, Lemmas, Theorems
Equational Proofs
Proof by Contradiction (*witness*)

Logic