Principles of Programming

Tuesday, 16 January 2018 1:19 PM

Program = data structure + algorithm

Software Engineering Cycle

1. Specification (contract)
2. Design
    - Modular programming, 1 task per module
    - Specify data flow among modules
    - Pre-/post- conditions, formal parameters
    - People who use programs must meet these pre-conditions
3. Verification
    - Assertion: Statements which are true (e.g. Conditions)
    - Invariants: loops
        - Capture correctness of algorithm
4. Coding: Translates design into a language
5. Testing: Designing test data
6. Refining the solution

Efficiency: choice of data structure

Modular Design Principles

- Abstraction and Information Hiding
    - Interfaces, access modifier keywords
    - Specifications: WHAT to do, not HOW (implementation)
- OO Design (inheritance hierarchy)
    - Objects encapsulate (combine) data and operations
    - Object = instance of class
    - Encapsulation, inheritance, polymorphism
- Top-Down Design (breaking into sub-problems)

Programming Issues

1. Modularity
    - Isolates errors and modifications
    - Eliminates redundancies
2. Modifiability
    - Use named constants
3. Ease of Use
4. Fail-Safe Programming: performs reasonably regardless of use
5. Style
    - Extensive use of methods
    - Private data fields
    - Error handling
    - Readability
    - Documentation
6. Debugging: use watches, assertions or print statements