

Chapter 1: Principles of Programming and Software Engineering

Data Abstraction & Problem Solving with C++ Fifth Edition by Frank M. Carrano



Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley. Ver. 5.0.

What is a Good Solution?

- A solution is good if:
 - The total cost it incurs over all phases of its life cycle is minimal
- The cost of a solution includes:
 - Computer resources that the program consumes
 - Difficulties encountered by users
 - Consequences of a program that does not behave correctly
- Programs must be well structured and documented
- Efficiency is one aspect of a solution's cost

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley. Ver. 5.0.

1-2

Key Issues in Programming

1. Modularity
2. Style
3. Modifiability
4. Ease of Use
5. Fail-safe programming
6. Debugging
7. Testing

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley. Ver. 5.0.

1-3

Key Issues in Programming: Modularity

- Modularity has a favorable impact on
 - Constructing programs
 - Debugging programs
 - Reading programs
 - Modifying programs
 - Eliminating redundant code

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley. Ver. 5.0.

1-4

Key Issues in Programming: Style

1. Use of private data members
2. Proper use of reference arguments
3. Proper use of methods
4. Avoidance of global variables in modules
5. Error handling
6. Readability
7. Documentation

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley, Ver. 5.0.

1-5

Key Issues in Programming: Modifiability

- Modifiability is easier through the use of
 - Named constants
 - The `typedef` statement

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley, Ver. 5.0.

1-6

Key Issues in Programming: Ease of Use

- In an interactive environment, the program should prompt the user for input in a clear manner
- A program should always echo its input
- The output should be well labeled and easy to read

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley, Ver. 5.0.

1-7

Key Issues in Programming: Fail-Safe Programming

- Fail-safe programs will perform reasonably no matter how anyone uses it
- Test for invalid input data and program logic errors
- Check invariants
- Enforce preconditions
- Check argument values

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley, Ver. 5.0.

1-8

Key Issues in Programming: Debugging

- Programmer must systematically check a program's logic to find where an error occurs
- Tools to use while debugging:
 - Single-stepping
 - Watches
 - Breakpoints
 - `cout` statements
 - Dump functions

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley, Ver. 5.0.

1-9

Key Issues in Programming: Testing

- Levels
 - Unit testing: Test methods, then classes
 - Integration testing: Test interactions among modules
 - System testing: Test entire program
 - Acceptance testing: Show that system complies with requirements

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley, Ver. 5.0.

1-10

Key Issues in Programming: Testing

- Types
 - Open-box (white-box or glass-box) testing
 - Test knowing the implementation
 - Test all lines of code (decision branches, etc.)
 - Closed-box (black-box or functional) testing
 - Test knowing only the specifications

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley, Ver. 5.0.

1-11

Key Issues in Programming: Testing

- Developing test data
 - Include boundary values
 - Need to know expected results

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley, Ver. 5.0.

1-12

Key Issues in Programming: Testing

- Techniques
 - `assert` statements to check invariants
 - Disable, but do not remove, code used for testing
 - `/*` and `*/`
 - Booleans
 - Macros

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley, Ver. 5.0.

1-13

Key Issues in Programming: Testing

- Stubs
 - An incompletely implemented method that simply acknowledges that it was called
- Drivers
 - A module that tests another module
 - For example, a `main` function

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley, Ver. 5.0.

1-14

Summary

- Software engineering
 - Techniques to facilitate development of programs
- Software life cycle
 - Phases through which software progresses, from conception to deployment to replacement to deletion
- Loop invariant
 - Property that is true before and after each iteration of a loop
- Evaluating the quality of a solution
 - Correctness, efficiency, development time, ease of use, cost of modification

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley, Ver. 5.0.

1-15

Summary

- Practice abstraction: Focus on what a module does, not how
 - For data-management problems
 - Encapsulate data with operations by forming classes
 - For algorithmic tasks
 - Break into subtasks
- UML is a modeling language used to express OO designs visually

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley, Ver. 5.0.

1-16

Summary

- A solution should be easy to modify
 - Modular
 - Independent of implementation of its modules
- Each function/method should be as independent as possible and perform one well-defined task
- Each function/method should include a comment: purpose, precondition, and postcondition
- A program should be as fail-safe as possible
- Effective use of available diagnostic aids is one of the keys to debugging
- Use “dump functions” to help to examine and debug the contents of data structures

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley, Ver. 5.0.

1-17