

Quotes from\*  
*Fundamentals of Computing II:  
Abstraction, Data Structures,  
and Large Software Systems*

Allen B. Tucker      Robert D. Cupper      W. James Bradley  
Richard G. Epstein      Charles F. Kelemen

McGraw-Hill, 1995

### The Craft of Programming, p. 1

[O]ne particular set of principles for program construction . . . , collectively called *function-oriented design*, [has] evolved over the last several decades. In function-oriented design, the designer focuses upon identifying the basic functions or processes that are required to solve a problem. . . . [T]he original problem is broken up into subproblems, and this decomposition process continues until each of the subproblems is simple enough to be solved by means of a single function.

### Six Principles for Program Design

**Problem Decomposition:** Find some basis for decomposing a problem into smaller and smaller subproblems.

**Formal Specifications:** Provide each function with a clearly stated precondition and a clearly stated postcondition.

**Functional Cohesion:** Each function should have a single purpose that can be explained with a simple sentence.

**Explicit Interfaces:** All data being passed into and out of a function must appear explicitly in the argument list of that function.

**Loose Coupling:** Avoid the use of global variables.

---

\*Selected by Tom Verhoeff, Eindhoven University of Technology, Department of Mathematics and Computing Science.

**Information Hiding:** The internal details of a function (e.g., local variables and the algorithm employed) should be hidden from the rest of the program.

### Using a Design Methodology: MAPS, pp. 10–11

The process of problem-solving is viewed in *MAPS*, a *Methodology for Algorithmic Problem Solving*, as consisting of a series of steps.

**Step 1: The dialogue**

**Step 2: The specifications**

**Step 3: The breakdown**

**Step 4: Defining abstractions**

**Step 5: Coding**

**Step 6: Testing and verification**

**Step 7: Presentation**

### Object-Oriented Design, pp. 53

[A]n alternative design methodology [is] *object-oriented design*. Object-oriented design focuses on the concepts inherent in the problem domain rather than the processes inherent in the problem solution. When a program is designed in this manner, its basic building blocks will be objects rather than functions.

### An Object-Oriented Design Methodology, pp. 54–56

An *object-oriented design methodology* has four major steps.

1. Identify the classes and objects that are part of the solution.
2. Identify and describe the data members that are contained within each type of object.
3. Identify each of the member functions that can act upon each type of object.
4. For each member function, describe its purpose, arguments, preconditions, and postconditions.

[C]lasses are designed with the intention of making them generally useful.