The science of computing

Computer science is a science of abstraction — creating the right model for a problem and devising the appropriate mechanizable techniques to solve it.
— A. Aho and J. Ullman

Objectives

After studying this chapter you should understand the following:

- the nature of software systems;
- how abstraction and decomposition are used to deal with complexity in problems;
- the fundamental components of a program, data and functionality;
- objects, and object-oriented systems and their relationship to software systems in general.
Objectives

- Also, you should be able to:
  - identify basic components in complex structures;
  - provide examples of abstraction levels from least abstract to more abstract;
  - explain how composition and abstraction simplify the organization of a system;
  - provide informal algorithms for common activities.

Nature of a software system

- A software system is a temporary solution to a changing problem…
- With two fundamental characteristics:
  - they are dynamic
  - they are complex.

Dealing with complexity: composition and abstraction

- Size of software system requires that …
  - It must be broken down into manageable pieces
  - It must be dealt with as a composite structure
  - Its parts must interact together
  - The more parts the more interaction
  - System complexity is proportional to the number of parts
Dealing with complexity: composition and abstraction

- **composition:**
  Process of building a system using simpler parts or components

- **abstraction:**
  Allows one to deal with system components with no worry about details of how components are constructed

- **abstraction ...**
  - Process of ignoring details irrelevant to problem at hand
  - Emphasizes essential ones
  - To abstract is to disregard certain differentiating details
Two aspects of a system: 

- **data** – information program deals with 
  - **data descriptions** are fixed, 
  - individual **data values** may change each time program runs

- **functionality** – what the program does with data 
  - **computation**: a goal-directed sequence of actions performed by a processor 
  - **algorithm**: set of instructions describing pattern of behavior guaranteed to achieve a goal

Object-oriented systems

- Use the Object Oriented approach for structuring systems 
  - Intended to produce systems that are ... 
    - composite 
    - modular 
    - built using abstraction 
    - organized around data
### Object-oriented systems

- Objects: abstractions used to describe the problem
- Functionality of system is distributed to the objects
- Each object has algorithms to accomplish specific tasks

### Summary

- Software systems are *complex*
- Software systems are *dynamic*
- To address these difficulties …
  - make extensive use of *abstraction*;
  - build systems that are *modular*, and *composite*.

### Summary

- To define a software system, we will include
  - a *description* of the *data* items to be manipulated by the system, and
  - a collection of *algorithms* that provide the system’s *functionality*
- We adopt an object-based methodology that structures the system around the data