CMSC 132: OBJECT-ORIENTED PROGRAMMING II

Collections

Department of Computer Science
University of Maryland, College Park
Collection

• Programs represent and manipulate abstractions (chunks of information)
  • Examples: roster of students, deck of cards
• One of the most universal abstractions is a collection
  • Represents an aggregation of multiple objects
  • Plus, perhaps, a relation between elements
  • Examples: list, set, ordered set, map, array, tree
• Supporting different operations
Data Structures

• Data structure
  • A way of representing & storing information

• Choice of data structure affects
  • Abstractions supported
  • Amount of storage required
  • Which operations can be efficiently performed

• Collections may be implemented using many different data structures
Java Collection Framework (JCF)

• Java provides several interfaces and classes for manipulating & organizing data
  • Example: List, Set, Map interfaces
• Java Collection Framework consists of
  • Interfaces
    • Abstract data types
  • Implementations
    • Reusable data structures
  • Algorithms
    • Reusable functionality
Collection Hierarchy

- Interface (red)
- Class (black)
Collection Interface


- Core operations
  - Add element
  - Remove element
  - Determine size (# of elements)
  - Iterate through all elements

- Additional operations supported by some collections
  - Find first element
  - Find $k^{th}$ element
  - Find largest element
  - Sort elements

- Collection vs. Collections
  - Collections is a class