5

Abstract Data Types

Modularity and Data Abstraction

5.1 Introduction and Motivation

modularity and information hiding

abstract data types

using Java's modularity features

ComparisonKey interface

5.2 Priority Queues—An Abstract Data Type

LEARNING OBJECTIVES

1. To learn what an abstract data type is by means of a simple example.
2. To understand how it is possible to replace the underlying representation for an abstract data type without changing the operations it presents to its external users through its interface.
3. To learn how to use information-hiding features of Java to hide the implementation details of an abstract data type's operations.

the priority queue ADT

A Priority Queue ADT Interface

```
/*
 *   The public interface for the PriorityQueue class contains
 *   the following method calls. Here, let PQ be a variable having
 *   a PriorityQueue object as its value, let X be a variable that
 *   contains a priority queue item, and let n be an integer variable.
 */

PQ = new PriorityQueue();        // creates an initially empty priority queue PQ

n = PQ.size();                          // returns the number of items in PQ and
                                            // stores it in the integer variable n

PQ.insert(X);                                                   // puts X into PQ

X = PQ.remove( );              // removes the highest priority item from PQ and
                                      // assigns it to be the value of the variable X
```

Program 5.1 Informal Interface for a PriorityQueue Class