

# 21 Custom Generic Data Structures

## Objectives

In this chapter you'll:

- Form linked data structures using references, self-referential classes, recursion and generics.
- Create and manipulate dynamic data structures, such as linked lists, queues, stacks and binary trees.
- Learn various important applications of linked data structures.
- Create reusable data structures with composition.
- Organize classes in packages to promote reuse.

## Outline

1. [21.1 Introduction](#)
2. [21.2 Self-Referential Classes](#)
3. [21.3 Dynamic Memory Allocation](#)
4. [21.4 Linked Lists](#)
  1. [21.4.1 Singly Linked Lists](#)
  2. [21.4.2 Implementing a Generic List Class](#)

- 3. [21.4.3 Generic Classes ListNode and List](#)
  - 4. [21.4.4 Class ListTest](#)
  - 5. [21.4.5 List Method insertAtFront](#)
  - 6. [21.4.6 List Method insertAtBack](#)
  - 7. [21.4.7 List Method removeFromFront](#)
  - 8. [21.4.8 List Method removeFromBack](#)
  - 9. [21.4.9 List Method print](#)
  - 10. [21.4.10 Creating Your Own Packages](#)
- 
- 5. [21.5 Stacks](#)
  - 6. [21.6 Queues](#)
  - 7. [21.7 Trees](#)
  - 8. [21.8 Wrap-Up](#)
- 
- 1. [Summary](#)
  - 2. [Self-Review Exercises](#)
  - 3. [Answers to Self-Review Exercises](#)
  - 4. [Exercises](#)
  - 5. [Special Section: Building Your Own Compiler](#)