

**1. Course number and name**

CptS 122: Data Structures C/C++

**2. Credits and contact hours**

4 credits, 3 lecture hours, 3 lab hours

**3. Instructor's or course coordinator's name**

Andy O'Fallon

**4. Textbook, title, author, and year**

P.J. Deitel and H.M. Deitel. 2017. *C++: How to Program* (10th ed.). Pearson Education, Inc.

*Other supplemental materials*

J.R. Hanly and E.B. Koffman. 2016. *Problem Solving & Program Design in C* (8th ed.), Pearson Education, Inc., Addison-Wesley.

P.J. Deitel and H.M. Deitel. 2016. *C: How to Program*, (8th ed.), Pearson Education, Inc. Prentice Hall.

**5. Specific course information**

- a. *Catalog description:* Advanced programming techniques: data structures, recursion, sorting and searching, and basics of algorithm analysis taught in C/C++ programming language.
- b. *Prerequisites or corequisites:* CPT S 121 with a C or better.

**6. Specific goals for the course**

By the end of the course, students will be able to

- Design, apply, and implement data structures including lists, stacks, queues, and binary trees (1a, 1b, 1d, 1e, 6a, 6b, 7b, 7d, 7f, 7g).
- Apply and implement several sorting algorithms (1a, 1b, 1d, 1e, 6a, 6b, 7b, 7d, 7f, 7g).
- Analyze algorithmic complexity (6c).
- Design, implement, test, and debug C++ programs applying modern tools and techniques (1a, 1b, 1d, 1e, 6a, 6b, 7b, 7d, 7f, 7g).
- Analyze a specification of a problem of moderate complexity, and construct a structured, elegant C++ program that solves the problem with the most appropriate data structure(s) (1a, 1b, 1d, 1e, 6a, 6b, 7b, 7d, 7f, 7g).
- Design and articulate solutions to lab problems with classmates (1a, 1b, 1d, 1e, 3a, 3b, 5b, 5g).
- Identify and implement test cases to edge scenarios in pseudocode and/or C++ code (6b, 6d).
- Identify, analyze, and solve C++ code and data structures interview questions in prep for internships (1a, 1b, 1c, 1d, 1e).

## **7. Brief list of topics to be covered**

- Data Structures
- Linked Lists
- Stacks
- Queues
- Binary Search Trees
- Recursion
- Software Design and Engineering Concepts
- Problem Solving Strategies
- C++ Classes and Objects
- Container Classes
- Value Classes
- UML Design Models
- Generic Classes
- Templates
- Operator Overloading
- Function Overloading
- Algorithmic Analysis (Big-O)
- Abstract Data types
- Inheritance
- Sorting Algorithms
- Polymorphism
- Intro to Graphics
- Exception Handling
- Standard Template Library
- Const, static, friend, pass-by-reference