Passaic County Community College
Academic Year: 2023-2024
Standard Syllabus

Department Chair: Merille Siegel

Course Code: CIS 260 **Course Title:** Data Structures

Department: CIS / Engineering **Semesters Offered:** Fall Day, Spring Evening

Course Description: This course introduces the knowledge required to create efficient, complex, and reliable software in the area of systems, applications, and games. Students will be introduced to implementation of Abstract Data Types (ADT's) and techniques of object-oriented analysis and design. Recursive methods and solutions for data structures including linked lists, stacks, queues, graphs, and trees are covered. In addition, various searching methods, sorting methods, and memory management techniques are covered. Students will design and implement programming assignments using an object-oriented programming language.
Prerequisites: CIS 161

Credits: 3 **Lecture Hours:** 3 **Lab/Studio Hours:** 0 **Clinical/Fieldwork Hours:** 0

Required Textbook/Materials:
Textbook: "Data Abstraction & Problem Solving with C++ : Walls and Mirrors"; Frank M. Carrano/ Timothy Henry, Pearson; 2013. ISBN-13: 978-0-13-292372-9

Reference: "Data Structures and Algorithms in C++" ; Adam Drozdek; Course Technology, Cengage Learning, 2013. ISBN-13: 978-1-133-60842-4

Special Facilities/Equipment:

Hardware: WinTel Platform PC or lab top.
Software: Any modern C++ compiler (**wxDev-C++** is a free integrated development environment (IDE) that is based on the popular Dev-C++.).
Storage Media: At least 1 GB Flash Drive for saving lab work.

Additional Time and Supplemental Requirements:
Based on a 15 week semester, students are expected to complete approximately 6 hours per week of assigned work outside of class.

- All assigned homework is completed out of class time. Students, who don't have a computer, can use the open lab to complete homework assignments. Students must read and enforce what they learned outside of the class.

Course Learning Outcomes:

Upon completion of this course, students will be able to:

1. Utilize classes, templates and UML in the design and analysis of ADT bags.

2. Apply memory management techniques to avoid memory leaks and dangling pointers.
3. Implement recursive and backtracking algorithms in problems solution.
4. Implement ADT bags using both static and dynamic data structures.
5. Apply data abstraction in specification of problems then implement the abstraction in constructing solutions.
6. Construct and quantify various storage structures such as (Stacks, Queues, Lists and Trees).

General Education Outcomes: This is not a general education course.

Grading Standards:

Activity	Contribution
Lab Projects (4 – 6)	30%
Tests/ Quizzes	35%
Final Exam	30%
Attendance/Activities/Etc.	5%

Course Content:

(Schedule and suggested topics, readings, and assignments subject to change based on instructor and instructional resource)

WEEK	TOPIC	Chapter
1	**Data Abstraction: The Walls** Object oriented Analysis and Design Achieving better solutions Designing Abstract Data Types(ADTs)	**1**
2	**C++ interlude 1– C++ Classes** Class definition and solution implementation Templates, Inheritance	**int 1**
3	**Recursion:** Recursive Solutions and examples, Recursion that returns a value, Recursion that performs an action, Recursion with arrays, organizing data.	**2**
4	**Array-Based Implementations:** The Approach Core Methods, Using Fixed-Size Arrays An Array-Based Implementation of the ADT Bag Defining, testing and Implementing the Core Methods Methods That Remove Entries Using Recursion in the Implementation	**3**
5	**C++ Interlude 2 Pointers, Polymorphism, and Memory Allocation**	**int 2**

- Memory Allocation for Variables and Early Binding of Methods
    - A Problem to Solve
- Pointers and the Program Free Store
    - Deallocating Memory
    - Avoiding Memory Leaks
    - Avoiding Dangling Pointers
- Virtual Methods and Polymorphism
- Dynamic Allocation of Arrays,
    - A Resizable Array-Based Bag.

WEEK	TOPIC	Chapter
6	**Link-Based Implementations**	**4**
	Preliminaries	
	The Class Node,	
	A Link-Based Implementation of the ADT Bag	
	The Header File	
	Defining the Core Methods	
	Implementing More Methods,	
	Using Recursion in Link-Based Implementations	
	Recursive Definitions of Methods in LinkedBag,	
	Comparing Array-Based and Link-Based Implementations	
7	**Recursion as a Problem-Solving Technique**	**5**
	Defining Languages	
	The Basics of Grammars,	
	Two Simple Languages,	
	Algebraic Expressions,	
	Kinds of Algebraic Expressions	
	Prefix/Postfix Expressions,	
	Fully Parenthesized Expressions	
	Backtracking	
	The Eight Queens Problem	
	The Relationship Between Recursion and Mathematical Induction	
	The Correctness of the Recursive Factorial Function,	
	The Cost of Towers of Hanoi	
7	**Stacks:**	**6**
	The Abstract Data Type Stack	
	Developing an ADT During the Design of a Solution	
	Specifications for the ADT Stack	
	Simple Uses of a Stack	
	Checking for Balanced Braces	
	Recognizing Strings in a Language	
	Using Stacks with Algebraic Expressions	
	Evaluating Postfix Expressions	
	Converting Infix Expressions to Equivalent Postfix Expressions	
	Using a Stack to Search a Flight Map	
	The Relationship Between Stacks and Recursion	
8	**Stack Implementations**	**7**
	An Array-Based Implementation	
	A Linked Implementation	
	Comparing Implementations	

WEEK	TOPIC	Chapter
8	**C++ Interlude 3 Exceptions**	**int 3**
	Background	
	A Problem to Solve	
	Assertions	
	Throwing Exceptions	
	Handling Exceptions	
	Multiple catch Blocks	
	Uncaught Exceptions	
	Programmer-Defined Exception Classes	
9	**Lists**	**8**
	Specifying the Abstract Data Type List	
	Using the List Operations	
9 + 10	**List Implementations**	**9**
	An Array-Based Implementation of the ADT List	
	The Header File	
	The Implementation File	
	A Linked Implementation of the ADT List	
	The Header File	
	The Implementation File	
	Using Recursion To Process a Linked Chain	
	Comparing Implementations	
11	**Queues**	**13**
	The ADT Queue	
	Simple Applications of the ADT Queue	
	Reading a string of characters	
	Recognizing Palindromes	
	A Linked Implementation of the ADT Queue	
	The Header File	
	The Implementation File	
	Comparing Implementations	
11 +12	**CIRCULAR QUEUES Priority Queues:**	**13**
	Implementation of circular queues and priority Queues.	
12 +13	**TREES:**	**15**
	Kinds of trees,	
	Height of trees	
	Full, Complete, and balanced binary trees.	
	The ADT Binary Tree	
	Representation, operations and traversal.	
13	**Binary search Trees:**	**15**
	Operations, construction, traversal.	

14	**Tree Implementation:** A linked based implementation.	**16**
15	**GRAPHS:** Terminology Nodes, edges, paths, Digraphs, Paths.	**20**
15	**Final – final project is also due.**	

<u>College Policies:</u>

For Information regarding:

- PCCC’s Academic Integrity Code
- Student Conduct Code
- Student Grade Appeal Process

Please refer to the PCCC Student Handbook and PCCC Catalog

<u>Panther Alert:</u>

The College will announce delayed openings, closings, and other emergency situations through the Panther Alert System. Students are encouraged to sign up for Panther Alert Notifications by logging into their student accounts through the PCCC website at www.pccc.edu and following Panther Alert System instructions.

<u>Notification for Students with Learnings Disabilities:</u>

If you have a disability, and believe you need accommodations in this class, please contact the Office of Accessibility Services at 973-684-6395, or email ods@pccc.edu. You should do so as soon as possible at the start of each semester. If you require testing accommodations, you must remind me (the instructor) one week in advance of each test.