BRONX COMMUNITY COLLEGE
of the City of New York
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

SYLLABUS: CSI 33 Data Structures 2 rec 2 lab 3 credits

PREREQUISITE: CSI 32 and CSI 35; and CUNY English Proficiency, or ENG 100 or 110, if required

TEXT: Data Structures and Algorithms Using Python and C++
by David M. Reed and John Zelle, Franklin Beedle and Assoc.

Goals of the course: To introduce students to working with data structures and algorithms as a way to develop solutions to various computational problems.

Objectives: To provide experience to students in using these skills:

1. Analysis of algorithms,
2. Class design, in Python and C++, based on performance requirements,
3. Understanding dynamic structures and their use in resource management, and
4. Correctly applying the fundamental searching and sorting algorithms.

Programming Projects: Students will complete 8-10 programming projects taken from the list of programming projects or comparable projects developed by the instructor.

Sections of the text	**Suggested exercises and projects**	
Chapter l: Abstraction and Analysis (½ week)		
1.2 Functional Abstraction	p. 33:1-10	
1.3 Algorithm analysis	p. 36:1,3,4,8	p.38:9
Chapter 2: Data Abstraction (1 week)		
2.2 Abstract Data Types	p.68:1-10	
2.3 ADTS and Objects	p. 71:1,2	p.71:1,3
2.4 An Examples ADT: Datasets		
2.5 An Example ADT: Rational		
Chapter 3: Container Classes (1 week)		
3.2 Python Lists	p.100:1-13	p.104:6,10
3.3 A Sequential Collection: A Deck of Cards	p.101:1,2,5,6,7	
3.4 A Sorted Collection: Hand		
3.5 Python List Implementation		
3.6 Python Dictionaries		

Chapter 4: Linked Structures and Iterations (1½ weeks)		
4.2 The Python Memory model	p.148:1-10	p.152:1,4
4.3 A linked Implementation of Lists	p.149:1,3	
4.4 Linked Implementation of a List ADT	p.151:1,2	
4.5 Iterators		
4.7 Lists vs. Arrays		
Chapter 5: Stacks and Queues (1 week)		
5.2 Stacks	p.181:1-10	p.184:1
5.3 Queues	p.182:1,2,5,6,7	
5.4 Queue Implementation	p.183:1,3	
5.5 An Examples Application: Queueing Simulations		
Chapter 6: Recursion (1 week)		
6.2 Recursive Definitions	p.212:1-10	p.215:5,7
6.3 Simple Recursive Examples	p.213:1,2,3	
6.4 Analyzing Recursion	p. 214:1	
6.5 Sorting		
6.6 A "Hard" Problem: The Tower of Hanoi		
Chapter 7: Trees (1½ weeks)		
7.2 Tree Terminology	p.245:1-10	p.248:1,3,4
7.3 An Example Application: Expression Trees	p.246:4,7,8	
7.4 Tree Representations	p.247:2,4,6	
7.5 An Application: A Binary Search Tree		
Chapter 8: C++ Introduction for Python (2 weeks)		
8.2 C++ History and Background	p.313:1-12	p.316:8
8.3 Comment, Blocks of Code, Identifiers, and Keywords		
8.4 Data Types and variable declarations	p.314:1,3,4	
8.5 Include Statements, Namespaces, and Input/Output		
8.6 Compiling	p.315:4,5,6	
8.7 Expressions and Operator Precedence		
8.8 Decision Statements		
8.9 Type Conversion		
8.10 Looping Statements		
8.11 Arrays		
8.12 Function Details		
8.13 Header Files and Inline Functions		
8.14 Assert Statements and Testing		

8.15 The Scope and Lifetime of Variables		
8.16 Common C++ Mistakes by Python Programmers		
Chapter 9: C++ Classes (½ week)		
9.1 Basic Syntax and Semantics	p.348:1-10	p.352:3
9.2 Strings	p.349:1,3,4,5	
9.3 File Input and Output	p.351:7	
9.4 Operator Overloading		
9.5 Class Variables and Methods		
Chapter 10: C++ Dynamic Memory (1 week)		
10.2 C++ Pointer	p.395:1-10	p.400:1
10.3 Dynamic Arrays	p.397:6,7	
10.4 Dynamic Memory Classes	p.399:3,4,5	
10.5 Dynamic Memory Errors		
Chapter 11: C++ Linked Structures (1 week)		
11.2 A C++ Linked Structure Class	p.422:1-5	p.424:1
11.3 A C++ Linked List	p.423:1,3,5	
11.4 C++ Linked Dynamic Memory Errors	p:424:1,2	
Chapter 12: C++ Templates (½ week)		
12.2 Template Functions	p.440:1-5	p.442:5
12.3 Template Classes	p.440:2,5, p.442:3	
Chapter 13: Heaps, Balanced Trees, and Hash Tables (1 week)		
13.2 Priority Queues and Heaps	p.478:1,2,7-10	p.483:2
13.5 Hash Tables	p.479:1,3,5, p.481:1	
Chapter 15: Algorithm Techniques (½ week)		
15.2 Divide and Conquer	p.546:1-5	
15.3 Greedy Algorithm	p.546:1	

Fall 2009 for Python/SEP/GL
August 2022 for prerequisite update EA