

## YORK TECHNICAL COLLEGE COURSE SYLLABUS

Course Prefix/Number:	<b>CPT 244</b>
Course Title:	<b>Data Structures</b>
Lec Hours/Week:	3.0
Lab Hour/Week:	0.0
Credit Hours/Semester:	3.0

[DL Attendance/VA Statement](#)  
[Textbook Information](#)

**Course Description** This course examines data structures widely used in programming. Topics include linked lists, stacks, queues, trees, and sorting and searching techniques.

### Course Competencies

Upon successful completion of this course, the student should be competent to complete the following tasks:

#### Algorithm Analysis Tools

- Compare growth rates of logarithmic, polynomial, and exponential functions
- Construct proofs (using mathematical induction and proof by contradiction)
- Use Sigma notation for sums, and determine the closed-form solution for a summation
- Use big-Oh, big-Omega, and big-Theta notation to describe the relative performance of algorithms
- Determine the worst case and best case time and space complexities for algorithms
- Explain the importance of average case analysis

#### Classic Data Structures

- Explain the uses and functionality of linked lists
- Discuss the costs and benefits of linked list implementations
- Explain the uses and functionality of doubly-linked lists
- Discuss the costs and benefits of doubly-linked lists
- Explain the uses and functionality of stacks
- Discuss the costs and benefits of stack implementations
- Explain the uses and functionality of queues
- Discuss the costs and benefits of queue implementations
- Explain the uses and functionality of binary and non-binary trees
- Discuss the costs and benefits of tree implementations
- Explain and execute preorder, postorder, and inorder tree traversals
- Implement at least two of the following data structures: linked list, doubly-linked list, stack, queue, binary tree
- Maintain a binary search tree: including insertion and removal of nodes
- Build and maintain heaps

#### Searching Methods

- Describe and implement a linear search
- Describe and implement a binary search
- Describe and implement hashing methods

#### Sorting Methods

- Describe the following n-squared algorithms: insertion, bubble, and selection sort
- Discuss the relative advantages and disadvantages of the insertion, bubble, and selection sort algorithms
- Describe the following sorting algorithms: shellsort, quicksort, mergesort, heapsort, binsort, radix sort
- Discuss the relative advantages and disadvantages of the shellsort, quicksort, mergesort, heapsort, binsort, radix sort algorithms

## Minimal Standards

Minimal standards of performance on all course competencies for receiving credit for the course are indicated by 60% accuracy on all evaluation instruments that address the course competencies listed above.

Recommended standards of performance on all course competencies for enrollment in subsequent higher-level computer technology courses are indicated by 70% accuracy on all evaluation instruments which address the course competencies listed above.

### Course Requirements

Students are responsible for attending all scheduled class meetings until they have completed all course requirements. Students are responsible for all material covered and for all assignments made in all classes. Students who are absent from a class more than 20% of the hours assigned will be withdrawn. Any student caught cheating or involved in other academic dishonesty will be given a grade of zero and will be subject to further disciplinary action.

### Evaluation Strategies/Grading Grading Scale

A test will be administered at the end of each module. A comprehensive exam will be left to the discretion of the course instructor. If a comprehensive exam is administered, it must not comprise more than 20% of the final grade.

90 – 100	A
80 – 89	B
70 – 79	C
60 – 69	D
Below 60	F

### Entry Level Skills

The student entering this course must have a good working knowledge of college algebra and C++.

Prerequisites: CPT 232 with a minimum grade of “C” and MAT 110

Co-requisites: CPT 233

### Topic/Content Outline

#### Module 1

##### Algorithm Analysis Tools

- A. Comparing the growth rates of logarithmic, polynomial and exponential functions
- B. Constructing proofs (using mathematical induction and proof by contradiction)
- C. Finding the closed-form solutions for summations
- D. Using big-Oh, big-Omega, and big-Theta notation to describe the relative performance of algorithms
- E. Determining the worst case, best case, and average case time and space complexities for algorithms

#### Module 2

##### Classic Data Structures

- A. Explaining the uses and functionality of linked lists
- B. Discussing the costs and benefits of linked list implementations
- C. Explaining the uses and functionality of doubly-linked lists
- D. Discussing the costs and benefits of doubly-linked list implementations
- E. Explaining the uses and functionality of stacks
- F. Discussing the costs and benefits of stack implementations
- G. Explaining the uses and functionality of queues
- H. Discussing the costs and benefits of queue implementations
- I. Explaining the uses and functionality of binary and non-binary trees
- J. Discussing the costs and benefits of tree implementations

- K. Implementing at least two of the following data structures: linked lists, doubly-linked lists, stacks, queues, and trees
- L. Executing preorder, postorder, and inorder tree traversals
- M. Implementing a binary tree data structure
- N. Implementing a binary search tree
- O. Inserting and removing of nodes from a binary search tree
- P. Building and maintaining heaps

### Module 3

#### Searching Methods

- A. Describing and implementing a linear search
- B. Describing and implementing a binary search
- C. Describing and implementing hashing methods

### Module 4

#### Sorting Methods

- A. Describing the implementations of the following n-squared algorithms: insertion, bubble, and selection sort
- B. Discussing the relative advantages and disadvantages of the insertion, bubble, and selection sort algorithms
- C. Describing the implementations of the following sorting algorithms: shellsort, quicksort, mergesort, heapsort, and radix sort
- D. Discussing the relative advantages and disadvantages of the shellsort, quicksort, mergesort, heapsort, and radix sort algorithms