BARTON COMMUNITY COLLEGE

**COURSE SYLLABUS**

## GENERAL COURSE INFORMATION

Course Number: PRGM 1020

Course Title: Data Structures and Algorithms

Credit Hours: 3

Prerequisites: 1025 C Programming, 1030 Java Programming, or permission of the instructor.

Division/Discipline: Workforce Training and Community Education

Course Description: This course is an intermediate level programming course which introduces students to common data structures and algorithms utilized in programming. Students are also introduced to algorithm analysis. Topics to be covered in this course include: lists, stacks, queues, trees, graphs, recursion, sorting/searching algorithms, and algorithm efficiency.

1. **CLASSROOM POLICY**

Students and faculty of Barton Community College constitute a special community engaged in the process of education. The college assumes that its students and faculty will demonstrate a code of personal honor that is based upon courtesy, integrity, common sense, and respect for others both within and outside the classroom.

The College reserves the right to suspend a student for conduct that is detrimental to the College’s educational endeavors as outlined in the College Catalog.

Plagiarism on any academic endeavors at Barton Community College will not be tolerated. Learn the rules or, and avoid instances of, intentional or unintentional plagiarism.

Anyone seeking an accommodation under provisions of the Americans with Disabilities Act should notify Student Support Service.

1. **COURSE AS VIEWED IN THE TOTAL CURRICULUM**

Data Structures and Algorithms is an intermediate level programming class. Students learn common data structures and algorithms used in program development which enhance problem solving abilities. It provides students with a solid foundation of computer programming, especially for those who are pursuing careers in computer science and technology.

The transferability of all college courses will vary among institutions, and perhaps even among departments, colleges, or programs within an institution. Institutional requirements may also change without prior notification. Students are responsible to obtain relevant information from intended transfer institutions to insure that the courses the student enrolls in are the most appropriate set of courses for the transfer program.

1. **ASSESSMENT OF STUDENT LEARNING/COURSE OUTCOMES**

Barton Community College assesses student learning at several levels:  institutional, program, degree and classroom.  The goal of these assessment activities is to improve student learning.  As a student in this course, you will participate in various assessment activities.  Results of these activities will be used to improve the content and delivery of Barton’s instructional program.

The student will be able to:

1. Recognize the proper use of the following data structures, algorithms and programming methods: Lists, Stacks, Queues, Trees, Graphs, Recursion, and Sorting & Searching algorithms
2. Apply the above data structures and programming methods to solve problems with a high-level programming language.
3. Compare alternative implementations of the above data structures.
4. Estimate the complexity of common algorithms.

# **COURSE COMPETENCIES**

The student will be able to:

1. Recognize the proper use of the following data structures, algorithms and programming methods: Lists, Stacks, Queues, Trees, Graphs, Recursion, and Sorting & Searching algorithms
2. Explain the purpose of an abstract data type
3. Describe a list and common list operations
4. Identify generic data types
5. Explain the purpose of a stack and its operations
6. Describe the purpose of a queue and its operations
7. Explain the purpose of a tree, common tree terminology, and operations
8. Identify variations of the tree
9. Describe the purpose of a graph and its terminology
10. Recognize common searching and sorting algorithms
11. Explain the use of hashing
12. Identity the components of a recursive algorithm
13. Apply the above data structures and programming methods to solve problems with a high-level programming language.
14. Apply a list in a program
15. Search and sort a list
16. Create a linked list
17. Implement a stack in a program
18. Apply a queue in a program
19. Implement a tree in a program
20. Traverse a tree and a graph
21. Apply a graph in a program
22. Trace simple recursive algorithms
23. Implement recursion in a program
24. Compare alternative implementations of the above data structures.
25. Compare array based vs. linked list implementations of lists, stacks, and queues
26. Estimate the complexity of common algorithms.
27. Identify notation and terminology of algorithmic complexity
28. Recognize levels of complexity in algorithms
29. Compare the complexity of common algorithms
30. **INSTRUCTOR EXPECTATIONS OF STUDENTS IN CLASS**
31. **TEXTBOOKS AND OTHER REQUIRED MATERIALS**

## REFERENCES

## METHODS OF INSTRUCTION AND EVALUATION

# **ATTENDANCE REQUIREMENTS**

## COURSE OUTLINE