

# Course Abstract

*If you need accommodations due to a disability, contact Disability Services in Edison Hall Room 100, 732.906.2546.*

*To foster a productive learning environment, the College requires that all students adhere to the Code of Student Conduct which is published in the college catalog and website.*

**Course ID and Name: CSC236 - Data Structures in Java**

**Department: Business and Computer Science**

Chairperson or Course Coordinator: Dr. Aslihan Cakmak

Office Location: ED123

E-mail Address: [ACakmak@Middlesexcc.edu](mailto:ACakmak@Middlesexcc.edu)

Telephone: 732-906-2526

**Prerequisites:** CSC-162 with a grade of "C" or better and MAT-129 or written permission of the department chairperson

**Co-requisites:** None

**Course Description:**

This course teaches different techniques of storing, accessing and processing data as utilized in the development of programs and algorithms. Topics include linked lists, stacks, queues, trees, and graphs. Algorithms for applications such as sorting, searching and merging will be analyzed and implemented. Solutions are designed using object-oriented techniques and implemented in the Java programming language. This course is the third course in the Java programming sequence and is a required course for students in the Computer Science Transfer Option.

**General Education Status:** N/A

**Credits:** 4

**Lecture Hours:** 3

**Lab Hours:** 3

**Learning Outcomes:**

**Upon successful completion of this course, a student will be able to:**

1. Develop programs using generic methods, classes and array based lists.
2. Create programs by applying the appropriate data structure: linked list, stack, and queue.
3. Design programs using binary trees, heaps, graphs, hashing, and collections.
4. Analyze the performance of various sorting and searching algorithms on a particular data structure.

## **Course Content Areas:**

1. ADTs, ArrayList, linked lists
2. Stacks, queues
3. Trees (binary, general, BST, tree traversals, B-tree, heaps, 2-3, 2-3-4, AVL, red-black, parse trees)
4. Sorting (insertion, selection, bubble sorts; merge sort, heap sort, quick sort; radix sort, shell sort, topological sort), binary search, sets, maps, hashing, shortest path algorithm and minimal spanning tree, Huffman coding/parse trees, graphs, bipartite