HOUSATONIC
COMMUNITY COLLEGE

Course Name: Data Structures

Course Number: CSC* E240

Credits: 3

Catalog description: Continues to develop discipline in program design and style especially as they relate to more complex problems. Topics include simple data structures, arrays, linked lists, queues, stacks and trees, string processing, and introduction to recursion.

Substantial lab work is required.

Prerequisite: CSC* E107 ***or*** CSC* E210

Corequisite or Parallel:

General Education Competencies Satisfied:

HCC General Education Requirement Designated Competency Attribute Code(s):

None

Additional CSCU General Education Requirements for CSCU Transfer Degree Programs:

None

Embedded Competency(ies):

None

Discipline-Specific Attribute Code(s):

☒ **COMP** **Computer Science Elective**

Course objectives:

General Education Goals and Outcomes:

None

Course Specific Objectives:

1. Write programs that demonstrate discipline in program design, style, expression, debugging and testing as they relate to larger programs than those expected in CS-115.
2. Develop clear algorithms.
3. Write programs that use basic aspects of string processing, recursion, and various data structures.
4. Demonstrate knowledge of various sorting techniques.
5. Compare sorting techniques with respect to efficiency.

Course Content:

CONSIDERATIONS OF PROGRAM DESIGN, TESTING AND DEBUGGING

A. Writing Detailed Specifications
B. Comparing Algorithms: The Big O Notation
C. Top-Down Design
D. Designing Data Structures
E. Implementing the Solution
F. Verifying, Testing and Debugging

ARRAYS AND RECORDS

A. One-Dimensional Arrays
B. Two-Dimensional Arrays
C. Records and Sets

POINTERS AND DYNAMIC MEMORY

A. The Syntax and Meaning of Pointer Variables
B. Using NEW, NIL and DISPOSE
C. Accessing Data through Pointers

VARIOUS REPRESENTATIONS OF STACKS AND QUEUES

A. Definition
B. The Logical Level of Stacks and Queues
C. The User Level
   1. Static Structures
   2. Linked Structures

LINEAR LIST

A. The Logical Level
B. The User Level
C. The Implementation Level
D. Operations on Linked Lists
E. Application Level

F. Circular Linked Lists

G. Doubly linked Lists

H. Linked Lists with Headers and Trailers

I. Implementation as an Array

RECURSION

A. Verifying Recursive Procedures and Functions

B. Writing Recursive Procedures and Functions

C. Using Recursion to Simplify Solutions

D. Recursive List Processing

E. How Recursion Works

1. Static Storage Allocation
2. Dynamic Storage Allocation
3. Issues Concerning Recursion and Storage

F. Removing Recursion

BINARY SEARCH TREES

A. The Logical Level

B. The Implementation Level

C. Recursive Binary Search Tree Operations

D. Binary Tree Traversals

E. Iterative Operations

F. Searching a Binary Search Tree

G. Binary Expression Trees

HEAPS (OPTIONAL)

SEARCHING AND SORTING TECHNIQUES

A. Sorting

1. Selection Sort
2. Bubble Ort
3. Quick Sort (Optional)
4. Heap Sort (Optional)

B. Searching

1. Linear Search
2. Binary Search
3. Hashing (Optional)

C. Efficiency Considerations

Date Course Created:

Date of Last Revision: 04/03/2017