Computer Science CSC 245
Data Structures

Credits and contact hours: Credit hours: 3, Contact Hours: 42 (1 contact hour = 50 minutes)

Instructor: Dr. Jodi Tims


Specific course information
a. Catalog description: This course focuses on the development of advanced data structures that are commonly used in computer science including stacks, queues, trees, and graphs. Emphasis is placed on the efficiency of the various data structures and the applications for which they are used. Advanced language features including inheritance, polymorphism, templates, and exception handling are also studied.
b. Prerequisites: CSC-240; Co-requisite: MTH-161
c. Required/Elective:
   i. Computer Science – required
   ii. Software Engineering - required

Specific goals for the course
a. Specific outcomes of instruction
   i. The student will demonstrate significantly more advanced skills in problem solving, algorithm analysis, and software engineering
   ii. The student will be able to define the term abstract data type and discuss how the concept is supported by the Java language
   iii. The student will have implemented advanced programs using ADTs in Java
   iv. The student will be able to describe various data structures and determine when each is applicable to a particular problem.
   v. The student will be able to explain what is meant by the term "efficiency" when discussing data structures and algorithms; determine the efficiency of various algorithms and data structures
   vi. The student will be able to explain the concept of exception handling and will have written code in Java that implements exception handling.

a. CAC Criterion 3 outcomes addressed by the course:
   a. An ability to apply knowledge of computing and mathematics appropriate to the program’s student outcomes and to the discipline
   b. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
   c. An ability to design, implement, and evaluate a computer-based system, process, component or program to meet desired needs
   d. An ability to function effectively on teams to accomplish a common goal.
i. An ability to use current techniques, skills, and tools necessary for computing practice.

j. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates the tradeoffs involved in design choices.

k. An ability to apply design and development principles in the construction of software systems of varying complexity.

c. EAC Criterion 3 outcomes addressed by the course:
   a. An ability to apply knowledge of mathematics, science, and engineering
   c. An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
   d. An ability to function on multidisciplinary teams.
   i. An recognition of the need for, an ability to engage in life-long learning.
   k. An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

Brief list of topics to be covered
   a. Review of introductory programming concepts (1 class period)
   b. Introduction to Java and Eclipse (2 class period)
   c. File I/O (2 class periods)
   d. Introduction to object orientation (3 class periods)
   e. Java classes (2 class periods)
   f. Introduction to dynamic memory (2 class periods)
   g. Linked lists (4 class periods)
   h. Stacks (2 class periods)
   i. Queues (2 class periods)
   j. Recursion (3 class periods)
   k. Binary trees, binary search trees, AVL trees, other trees (8 class periods)
   l. Hash tables (2 class periods)
   m. Performance evaluation (2 class periods)
   n. Graphs (3 class periods)
   o. Exams (4 class periods)