

COURSE DESCRIPTION

Course No.	CSC230	Course Title	Elementary Data Structures and Algorithms
Course Type	Required		
Sem. Hours	3	Coordinator	Lydia Fritz

Current Catalog Description:

Advanced syntax of high level language taught in CSC130. Emphasis on modularization and abstraction. Big-O analysis of algorithms. Design and use of abstract data types with various implementations.

Textbook:

Data Structures and Abstractions with Java, 4th edition. F. Carrano & T. Henry. Pearson. 2015.

References:

None

Course Outcomes:

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

1. Apply the object oriented programming concepts object composition, inheritance and polymorphism
2. Design abstract data types and interfaces
3. Implement sequential abstract data types using both array and linked data
4. Describe characteristics of elementary searching and sorting algorithms
5. Implement recursive solutions to common computing problems
6. Evaluate algorithm performance using asymptotic analysis

Activities Enabling Program Outcomes (POx refers to program student outcome x)

Instruction: In this course, students are introduced to the following topics:

- Creation of classes from other classes, using the concepts of composition and inheritance. (POk, POc, POj)
- UML to illustrate a software system's necessary classes and their relationships. (POc, POj, POk)
- Abstract Data Types as a means for conceptually defining a sequential data structure. (POk)
- Sequential data structures implemented with array and linked data. (POa, POk)

- Selection of an appropriate data structure for a particular problem. (POj, POk)
- Analysis of iterative and recursive sorting algorithms. (POa)
- Design and analysis of recursive algorithms. (POa, POB, POj)
- Calculating Big-Oh, Big-Omega and Big-Theta bounds on running time for particular algorithms. (POa, POB, POi)

Student Activities and Assessment: Every offering of this course will include (details of assessment criteria and expectations are in outcome rubrics):

- One or more programs/assignments in which students choose and use computing and mathematical principles, and translate this into a working program (POa)
- One or more test questions that require students to properly use and interpret computing and mathematical terms (POa)
- One or more test questions that require students to evaluate computational solutions (POa)
- One or more test questions that require students to derive and analyze time or other resource complexity (POj)
- One or more programs/assignments in which students practice appropriate design and development principles (POk)

Prerequisites by Topic:

Students must have
<ul style="list-style-type: none"> • a grade of at least C (2.0) in CSC 130 (Introduction to Computer Science)

Major Topics Covered in the Course:

<ul style="list-style-type: none"> • The programming process, algorithm analysis and design • Object design techniques, including object composition and inheritance • Sorting and searching algorithm design and analysis • Making algorithms generic using generic data types • Problem solving using recursion and the analysis of recursive algorithms • Analysis of various data structures, including dynamic arrays, 2-D arrays, stack, and queue • Polymorphism

Estimated Curriculum Category Content (Semester hours):

<i>Area</i>	<i>Core</i>	<i>Advanced</i>	<i>Area</i>	<i>Core</i>	<i>Advanced</i>
Algorithms	1	0	Software design	.5	0
Data structures	1.5	0	Prog. Languages	0	0
Comp Org & Arch	0	0			