CEdu 582 – Data Structures

Department of Educational Computing
Cardinal Stritch University College of Education
MSIT III Oct 27 – Dec 1

Instructor:

Instructor: James (Jim) Kasum, Ph.D.
Email: jkkasum@gmail.com
Phone: 414-410-4021 (s) or 262-238-9870 (h)
Office Hours: By appointment
Class Time and Location: M 5:00 PM in BH 038

Credit Hours: 3 credits

Course Description

Data Structures is an essential foundation for anyone who programs or is curious about how software works. This course deals with data management concepts, data storage and data access techniques. Course topics include abstract data types, encapsulation, strings, arrays, linked lists, file structures, recursion, graphs, trees, sorting, searching and an introduction to object oriented programming concepts.

Program Outcomes

- Be competent practitioners, able to apply educational and technology-related research, the psychology of learning, and instructional design principles in creating an action research plan which connects current research to practice.
- Be effective problem solvers, able to integrate technology into curricula effectively in a culturally diverse and pluralistic society.
- Be competent practitioners, able to use technology and take a leadership role in its implementation.
- Be able to identify the dynamics of change when the restructuring of curriculum takes place.

The program outcomes are tied to Standards 4, 5, 7, & 10 of the Wisconsin Standards for Teacher Development and Licensure:

- The teacher understands and uses a variety of instructional strategies, including the use of technology to encourage children's development of critical thinking, problem solving, and performance skills.
- The teacher uses an understanding of individual and group motivation and behavior to create a learning environment that encourages positive social interaction, active engagement in learning, and self-motivation**.**
- The teacher organizes and plans systematic instruction based upon knowledge of subject matter, pupils, the community, and curriculum goals.

- The teacher fosters relationships with school colleagues, parents, and agencies in the larger community to support pupil learning and well being and who acts with integrity, fairness and in an ethical manner**.**

The program outcomes are also tied to Standards 2, 3, & 4 of the Wisconsin Standards for Administrator Development and Licensure:

- The administrator leads by facilitating the development, articulation, implementation, and stewardship of a vision of learning that is shared by the school community.
- The administrator manages by advocating, nurturing and sustaining a school culture and instructional program conducive to pupil learning and staff professional growth.
- The administrator ensures management of the organization, operations, finances, and resources for a safe, efficient, and effective learning environment.

Measurable Course Objectives

- **Course Objective 1:**
  Students will be able to **explain** the concept of a data structure, its representation an abstract data type and the concept of data encapsulation.
  Assessment:
  Students will participate in class discussion and take two written examinations covering this material.
  Criteria:
  Students will score at least 80% on the written examination.

- **Course Objective 2:**
  Students will be able to **explain and identify** the data structures: String, Stack, Queue, Linked List, and Binary Trees, using a current high level programming language.
  Assessment:
  Students will participate in class discussion and take two written examinations covering this material.
  Criteria:
  Students will score at least 80% on the written examination.

- **Course Objective 3:**
  Students will be able to **design and implement** programming applications which incorporate in an integral way appropriate data structures Assessed by: 1, 2, 3
  Assessment:
  Students will participate in class discussion and take two written examinations covering this material.
  Criteria:
  Students will score at least 80% on the written examination.
  Assessment:
  Students complete assigned programming projects that provide an opportunity to implement the conceptual data structures within the context of a substantial application.
  Criteria:
  Programming projects will be graded on a 100-point basis using the program evaluation rubric attached to the end of the syllabus.

- **Course Objective 4:**
  Students will be able to explain and **implement** standard sorting, searching and file processing techniques
  Students will be able to **design and implement** programming applications which incorporate in an integral way appropriate data structures Assessed by: 1, 2, 3
  Assessment:
  Students will participate in class discussion and take two written examinations covering this material.
  Criteria:
  Students will score at least 80% on the written examination.
  Assessment:
  Students complete assigned programming projects that provide an opportunity to implement the conceptual data structures within the context of a substantial application.
  Criteria:
  Programming projects will be graded on a 100 point basis using a program evaluation rubric as described in objective 3.

Grading Scheme or Weight Attached to Assessments

Assignment/Assessments	Total Weight
Midterm Examination	20%
Programming Assignments (4 total)	60% (15% each)
Final Examination	20%

Grading Guidelines

Objective Tasks

93-100%	A	91-92%	A-				
88-90%	B+	85-87%	B	82-84%	B-		
78-81%	C+	75-77%	C	70-74%	C-		
68-71%	D+	65-67%	D	60-64%	D-	below 60	F

Projects and Assignments, Grade Definitions: (See rubric in Course Objective 3)

A: Rubric was followed completely and accurately; the assignment was handed in on time, neatly presented, and well organized.

B: Rubric was followed completely and was generally accurate; the assignment was handed in on time, neatly presented and well organized.

C: Rubric was generally complete and generally accurate, the assignment was not handed in on time. There was some attention to neatness and organization.

Recommended Text and Other Resources

There is no required text for this course, however students may wish to have a good programming text as a language reference and/or a standard data structures text.

How To Program (Sixth Edition), by H. M. Deitel & P. J. Deitel, ISBN: 013038478X, Prentice Hall, 2005.
There are a large number of print resources available in the University library which deal with Data Structures, Visual Basic, C++ or the Java language. You are encouraged to study and research these as needed.

General Course Expectations

- It is crucial that students attend all classes and study team meetings. If an absence is deemed necessary, this will be handled on an individual basis.
- Active class is an important component.
- Word processed work is expected unless otherwise indicated.
- All work is to be in the student's own words unless quotation marks and referenced pages are provided. Use APA 5th edition guidelines.
- APA, 5th edition, style should be used for documentation of references.
- It is expected that work will be turned in on the assigned date. If there is a problem, please contact the instructor before the due date.

University Policies:

Statement of Academic Integrity

Inherent in the mission of Cardinal Stritch University is the strong belief in the principle of academic integrity. Student's actions reflect their moral character and, by extension, the University's reputation. Therefore, all students are expected to recognize and to abide by the policy on academic integrity found in the student handbook.

Statement of compliance with the Rehabilitation Act of 1973

Cardinal Stritch University and this instructor wish to positively affirm the intent of the American Disability Act. Any person enrolling in this course who may require alternative instructional and/or evaluative procedures due to a disability should feel free to discuss these needs with the instructors so that appropriate arrangements can be made.

Policy on Attendance

Attendance at all class meetings is mandatory. Students are expected to attend all scheduled classes and study team meetings. The following apply to all enrolled students:

Promptness:
Prompt arrival at each class is critical because of the limited time available to conduct in-class activities during the course. Instructors may take actions they deem appropriate if consistent tardiness is observed.

Absence:
Any absence from a scheduled class requires that the student notify the instructor. The student is required to make up all "missed" work as determined by the instructor, by the next class meeting.

Policy on Late or Missing Assignments

All course assignments should be completed by the due date. Work that is submitted after the date due is subject to penalty as described in the program grading rubric in Objective 3.

Sequence of Classes, Topics and/or Content

Class Date:	**Topic or Activity, Readings, Due Dates:**
Oct 27	Concept of a Data Structure and an abstract data type. Why study Data Structures? Content: Postfix arithmetic notation, Stack concepts and implementation, Inventing a calculator using stacks, Analysis and implementation of calculator project.
Nov 3	Completion of postfix notation discussion. Postfix assignment Queues, queue tools Uses of queues in applications.
Nov 10	Queues tool implementations Uses of queues in applications. Linked List concepts, linked list tools. Logical order versus physical order, Pointers Linked list implementations and applications
Nov 17	Dynamic memory allocation, pointer types Dynamic stack assignment Recursion – something for nothing ? Recursive techniques and discussion of examples
Nov 24	Binary trees, an incredibly useful structure used in data bases and other applications Sorting and searching techniques.
Dec 1	Sorting and searching techniques continued Exchange, insertion and selection sorts. Course Wrap-Up. Hooray, we've made it ! Final Exam.

Name: ______________________________ Assignment #_____

Essential Elements Check List:

___ Complete hard copy of source code and commentary.
___ Complete source code on disk
___ Executable code on disk
___ Cover sheet hard copy with all assignment elements in folder

Score	**17**	**12**	**7**	**0**
	User Interface is clean and designed for efficient use, containing complete directions for use as needed	User Interface is satisfactory but could be more efficient in design or user directions could be more complete	User interface is incomplete, missing features, confusing or lacks sufficient user directions	User interface is missing major required features or complete lack of necessary user directions
	Algorithm employed provides an efficient solution to the problem	Algorithm employed provides a solution to the problem but could be more efficient	Algorithm employed provides a partial but not complete solution to the problem	Algorithm does not solve the problem
	Program design is complete and efficient with proper use of parameters, variables and program structure	Program design is adequate but could use variables or parameters more efficiently	Program design is incorrect or incomplete. Not all required features are accounted for.	Program design does not implement required major program features
	Program documentation is complete	Program documentation is sufficient but could be more complete	Program documentation is lacking in quantity and/or quality	Program documentation is largely or totally missing
	Program execution is completely correct	Program execution is sufficient but has minor glitches or quirks	Program execution is correct for some program features but not all	Program execution fails to correct execute multiple major program features
	Assignment Package is complete with cover sheet, commentary, hard copies and disk files	Assignment package lacks one or two items	Assignment package lacks more than two items	Assignment package not submitted
Total	**Maximum 100 points** **Point Total Interpretation:** A 91-100 A- 90 B+ 89 B 81-88 B- 80 C+ 79 C 71-78 C- 70 D+ 69 D 61-68 D- 60 F 0-59			