



Curriculum Vita August 2019

Name: R. Daniel Creider, Ph.D., Associate Professor
Computer Science and Information systems
Texas A&M University-Commerce
PO Box 3011
Commerce, TX 75429-3011

Office: Jour 216

Office Phone: (903) 886-5407

Email: Daniel.Creider@tamuc.edu

Academic Degrees

Ph.D., Baylor University, Waco, Texas; Major - Experimental Psychology, 1978
M.S., East Texas State University (now TAMUC), Commerce, Texas; Major - Computer Science, 1979
M.S., Auburn University, Auburn, Alabama; Major - General/Experimental Psychology, 1971
B.S., Southwest Missouri State University, Springfield, Missouri; Major - Psychology, 1969
B.A., Central Bible College, Springfield, Missouri; Major - Bible/Music, 1968

Teaching Experience

August 1984 to present, Associate Professor
August 1978 to August 1984, Assistant Professor
Texas A&M University-Commerce, Computer Science and Information Systems Department, Commerce, TX.;

August 1977 to August 1978, graduate assistant, Baylor University, Computer Center Waco, Texas
August 1974 to August 1977, graduate assistant, Baylor University, Psychology Department, Waco, Texas

Publications (refereed)

Creider, R. Daniel, Ammouri, Emad, and Ayad, Ali. "An Intelligent Advisor for the Immigration and Naturalization Service". *Proceedings of the ISCA Fourth Golden West International Conference on Intelligent Systems*, June 12-14, 1995, pp. 239-243.

Creider, Daniel., Legge, Gaynor. A computer Science Masters Degree Curriculum at a Small University. *The Journal of Computing in Small Colleges*. Vol 12, No. 4. March 1997. pp 31-39.

Creider, R. Daniel, Palmer, Michael, Matrix Manipulations in C/C++ Using Dynamically Allocated Memory. *Proceedings of The Fifth World Conference on Integrated Design and Process Technology*, IDPT Vol. 1, pp. 21 (abstract), 1999-2000. Complete paper on CD.

Creider, R. Daniel, Sundar Singh, Paul S. "Differential Diagnosis of Allergic Rhinitis and Sinusitis An Expert System". *Proceedings of The Second World Conference on Integrated Design and Process Technology*, IDPT-Vol. 2, pp. 251-254, 1996.

Kolstad, K. K., Coker, Donald R., and Creider, R.D. "The Improvement of Testing Through Cooperation Between Teachers and Computer Personnel". *Educational Technology*, Vol. 24, No. 5, May 1984.

Kolstad, K. K., Creider, R.D., Kolstad, R. A. "Structural Cueing on MULTIPLE-CHOICE Test Items". *Education*, Vol. 107, No. 3, Spring 1987.

Kolstad, R., Creider, R.D., Cassel, R.N. "Use of the Democratic Maturity Test to Predict Freshman College Dropouts". *Scientia Paedagogica Experimentalis*, Vol. XXXVI, pp. 263-266, 1999.

Myers, S.M., Creider, R. Daniel. "Aero Expert: An Electrical Design Expert System". *Proceedings of the ISCA Fourth Golden West International Conference on Intelligent Systems*, June 12-14, 1995, pp. 234-238.

Suh, Sang C., Creider, R. Daniel, Kandula, Veerasekhar. "Prospective View on Intelligent Databases." *Proceedings of the Third Golden West International Conference on Intelligent Systems*. Las Vegas, June 2-4, 1994, pp. 757-761.

Presentations

- Creider, R. Daniel, Palmer, Michael, Matrix Manipulations in C/C++ Using Dynamically Allocated Memory. The Fifth World Conference on Integrated Design and Process Technology, Dallas Texas June 4-8, 2000
- Creider, R. Daniel., Ammouri, Emad., and Ayad, Ali. An Intelligent Advisor for the Immigration and Naturalization Service. Fourth Golden West International Conference on Intelligent Systems. San Francisco, June 12-14, 1995
- Creider, Daniel., Legge, Gaynor. A computer Science Masters Degree Curriculum at a Small University. Consortium for Computing in Small Colleges: Eighth Annual South Central Conference; April 11-12, 1997, San Antonio, Texas.
- Creider, R. Daniel., Sundar Singh, Paul S. Differential Diagnosis of Allergic Rhinitis and Sinusitis An Expert System. The Second World Conference on Integrated Design and Process Technology, Austin Texas December 1-4, 1996.
- Myers, S.M., Creider, R.D., Aero Expert: An Electrical Design Expert System. Fourth Golden West International Conference on Intelligent Systems. San Francisco, June 12-14, 1995
- Suh, Sang C., Creider, R. Daniel, Kandual, Veerasekhar. Prospective View on Intelligent Databases. Third Golden West International Conference on Intelligent Systems. Las Vegas, June 1994.

Presentations Supervised

Supervised graduate student presentation (Divya Kandala) for the 6th Annual Texas A&M University System Pathways Student Research Symposium, Nov 8, 2008, Commerce, TX. Topic was Programming Performance. Student won second place for Master's level in Computer Science

Supervised graduate student presentation (Joythi Pamulapati) for the 7th Annual Texas A&M University System Pathways Student Research Symposium, Nov 13-14, 2009, Laredo, TX. Topic was Programming Efficiency.

Supervised graduate student presentation (Song Huang) for the 3rd annual Federation Graduate Student Research Symposium on April 27, 2012 at Texas Woman's University, Denton, TX. Topic was **ProOSC** (Programmer Optimized Source Code) Programming Method - A Method of Programming for Faster Execution.

Research – Funded

Two HP servers (two 16 core servers, DL3-386, each with 512 GB RAM and 14TB storage) were donated in November 2018 by Apple Inc. to be used in the prime number research. In a separate shipment Apple Inc. also donated new disk drives to be installed in both of the servers.

Contract research funded by E-Systems, Greenville Division examining the use of Artificial Intelligence in the defense industry for the years given below. CO-PI with Dr. Richard Detmer, (also in Computer Science Dept.)

June 1983 through December 1983 in the amount of\$30,538.

April 1984 through December 1984 in the amount of\$48,402.

January 1985 through May 1986 in the amount of\$31,740.

Awards

Spring 2011 Student Recognition Award for Teaching Excellence

Awarded by The Texas A&M University System July 15, 2011; Award amount \$2,500

Fall 2010 Student Recognition Award for Teaching Excellence

Awarded by The Texas A&M University System April 15, 2011; Award amount \$2,500

Research Activities

Created the **ProOSC (Programmer Optimized Source Code) Programming Method**. This is a method of implementing the algorithm in code in such a way so as to eliminate unnecessary memory, eliminate unnecessary statements (code operations), eliminate redundant operations, and other modification which may be unique to a specific algorithm to improve the speed of the program and reduce memory requirements. This method of programming cannot be duplicated by an optimizing compiler. A compiler cannot modify the coded algorithm. A compiler cannot eliminate a termination test for a loop control and it cannot identify redundant operations. The object of this method is not to write a program and then make it efficient by systematically going through the code line by line looking for ways to change the code to make it more efficient. The object is to write efficient code while

converting the algorithm into the correct optimized syntax for a specific high level language. This method requires a change in the programming techniques used by a programmer, i.e. a paradigm shift in programming techniques. This is a method which is not taught in beginning programming classes and is not taught by anyone else.

The technique of optimizing the source code by reducing the number of operations while implementing the algorithm has been examined in the following algorithms. This is a partial list which continues to expand.

- Sequentail search – improved by combining the value test and the end of list test
- Frequency count in an unordered list – improved by reducing the number of times the end of list test is executed
- Frequency count in an ordered list – improved by using a binary search to find the value and then a modified sequential search to find the first and last occurrence; end of list test eliminated in the sequential search part of the algorithm.
- Frequency distribution using two arrays (unique value array and count array) – improved by eliminating the end of list test in the array containing the unique values
- Frequency distribution using one array (frequency count) – improved by eliminating the end of list while count the occurances of each value
- Mode computation (unordered data no extra arrays) – improved by reducing the number of times the end of list test is executed.
- Mode computation (unordered data) using three different algorithms – one requiring no extra memory, one requiring 1 additional array and the last one requiring 2 additional arrays. Each algorithm is implemented in a way to reduce the number of operations.
- Mode computation using ordered data – improved by performing a end of list test only when the successive vauue in the data set changes.
- Additional algorithm that have been examined
 - Comparison of arrays
 - Set operations – union, intersection and symantic difference
 - Removing all occurances of a specific value
 - Removing all occurances of a specific value except for the first occurrence
- Data structures and algorithms on data structures that have been modified
 - Linked lists implemented with an efficient node in place of a dummy node
 - Non-Fradmented Array Based Linked List
 - Insertion sort
 - Tag sort

Created a new implementation of the array based linked list called the Non-Fragmented Array Based Linked List (NFABLL). The NFABLL structure does not intermingle the active nodes with the empty nodes in the linked list and results in a number of advantages not possible in the usual implementation. Future research on this structure involves developing an algorithm to perform a binary search on this structure while maintaining the advantages of a linked list.

Current research involves identifying patterns in prime numbers (numbers in general) and developing algorithms to identify/find successive larger consecutive prime numbers. An algorithm was developed in 2017 and implemented in 2018 to store all potential prime numbers in an infinite bit table. A single binary digit is used to represent each potential prime number (no actual numeric value is stored in the table); if the bit is one the number represented is a prime otherwise the bit is zero designating a composite number. The position of the bit in the table is used to reconstitute/reconstruct the actual number represented in the table. Segments of the table are constructed separately.

Segments of the bit table have been created which currently contains 52,794,898,134,631 prime numbers and includes 2,046,461,318,833 prime pairs. Additional external storage will be required to further extend the bit table. The archive of consecutive prime numbers will be used to evaluate the distribution of prime numbers and search for patterns.

Honors Thesis Student

Walker, Grady M., *Is the Josephson Effect the Answer to the Supercomputer?*, December 1981

Young, Preston P., *Improving Speed Through Programmer Optimized Source Code*, May 2016

Thesis Students (committee chair, 20)

- Mosley, Burt, *Developing A Generic Expert System for Solving Configuration Problems*, May 1985
Rhee, Rhillkyu, *A Syntactic Pattern Recognition System for Handwritten Characters*, August 1986
Blair, Richard N., *Configuration Management in Symbolic Processing Environments*, May 1987
Burks, Mike W., *User Interfaces and User Interface Management Systems: A Survey*, May 1988
Campbell, K. Scott, *Knowledge Acquisition in a Knowledge-Based Intelligence Analysis System*, December 1988
Moquin, Bertrand P., *Software Configuration Management an Expert System Candidate*, December 1988
Norris, David G., *An Expert System to Calculate Channel Capacity According to Manning's Equation*, May 1990
Slaughter, Stephen R., *An Application of Artificial Neural Systems in Trend Analysis*, May 1990
Wilcoxson, John K., *Application of Sublanguages in Natural Language Translation*, May 1990
Portwood, Marion M., *Rule Adaptation and Modification in Machine Learning*, August 1990
Weatherman, Wesley L., *Expert Systems Design and Implementation of XL.PRL*, December 1990
Hoover, Lucian T., *EEG Monitor Software*, May 1992
Benge, Carey S., *NFS (Network File System) A Study of the Functionality, Design , and Implementation of NFS in a UNIX Based Environment*, December 1992
Skauge, Suzanne E., *Design and Implementation of Computerized Test Administration Software*, May 1993
Coco, Roger K., *Expert System Shell Design*, August 1993
French, Lonnie G., *Fossil Expert System*, May 1994
Khamitkar, Charudatta N., *Universal Healthcare System on the Internet*, December 1996
White, Pablo, *The Root Word Analysis of the English Language*, May 2004
McCord, William Keith, *Pathfinding Algorithms in Two-Dimensional Computer Games Using C#*, May 2005
Huang, Song, *Large Number Multiplication: A Comparison of Single Processor and Multiple Processors Implementation*, August 2013

Courses Taught

- Texas A&M University-Commerce (courses in parentheses are no longer offered)
- 126 Introduction to Computing
 - 152 Programming Fundamentals II
 - (236) Introductory FORTRAN Programming
 - 241 Assembly Language Programming
 - 415 GLB/Info Sec Law Ethics (previous number was 251)
 - (337) Advanced FORTRAN Programming
 - 380 Web Programming and Interface Design
 - 502 Internship in College Teaching of Computer Science
 - 504 Introduction to Computer Application
 - 506 Introduction to Computer-Oriented Research Applications
 - 515 Fundamentals of Programming
 - 520 Information Structure and Algorithms Analysis
 - 526 Database Systems
 - 537 Advanced Concepts of Programming Languages
 - 538 Algorithms and Programming Techniques for Artificial Intelligence Applications
 - 539 Expert Systems
 - 597 Neural Computing
 - 597 Programming Performance