

# Charles D. Allison

Curriculum Vitae

## Education

- **M.S.** Applied Mathematics, *University of Arizona* 1985  
(ABD toward doctorate)  
*Emphasis:* Computer Science  
*Research:* Automatic Numerical Stability Analysis of Matrix Algorithms
- **M.S.** Mathematics, *Brigham Young University* 1978  
*Minor:* Statistics  
*Thesis:* “Floquet Theory via Functional Equations”
- **B.S.** Mathematics, *Brigham Young University* 1976  
*Minor:* Portuguese

## Awards

- *Award of Excellence for Scholarship*, College of Engineering and Technology, Utah Valley University, 2019
- *Outstanding Educator of the Year*, Utah Valley University, 2016, 2013, 2004
- *Board of Trustees Award of Excellence*, Utah Valley University, 2010
- *Faculty Excellence Award*, Utah Valley State College, 2004
- *NASA Ph.D. Traineeship in Applied Mathematics*, Univ. of Arizona, 1981 – 1983

## Memberships (past and present)

- Association of Computing Machinery (Senior Member)
- Sigma Xi, The Scientific Research Society
- Society for Industrial and Applied Mathematics
- American Mathematical Society
- Mathematical Association of America
- Pi Mu Epsilon, The Honorary National Mathematics Society

## Teaching Experience

### Parkland College, Champaign, IL

*Adjunct Professor*, Dept. of Mathematics and Natural Science, August 2023 –.

### Utah Valley University, Orem, UT

*Professor Emeritus*, Dept. of Computer Science, July 2022 – present.

*Professor*, Dept. of Computer Science, July 2011 – June 2022.

*Chair*: Dept. of Computer Science, January 2013 – 2016.

*Associate Professor*, Dept. of Computing & Networking Sciences, July 2006 – June 2011.

*Assistant Professor*, Dept. of Computing & Networking Sciences, August 2001 – June 2006.

Teaching Computation Theory, Discrete Structures, C++ Software Development, Numerical Software Engineering, Design Patterns, Analysis of Programming Languages, Python, D, Rust, and Java ▪ Faculty Senate President (2008-2010) ▪ UVSELF (Utah Valley Senior Executive Leadership Forum) Fellow, 2008-2009 Cohort ▪ Member of the Advisory Board for the Center for Engaged Learning (2008-2010) ▪ Served on UVU's Scholarly and Creative Activities Council (Undergraduate Research), 2005 – 2006 ▪ Faculty senate representative (2004-2007) ▪ Program Coordinator of CS emphasis (2001-2012) ▪ Member of the Math Task Force Committee ▪ Member of CS Department Master's Degree Curriculum Committee ▪ Member of University Faculty Workload Conference Committee ▪ Member of Presidential Search Committee (2008-2009) ▪ Member of University Accreditation Governance Sub-committee (2009-2010) ▪ Developed a Relational Algebra Engine for use in a Database Theory course.

### **Pima Community College, Tucson, AZ**

*Instructor*, Department of Computer Science, 1983 – 1984 (full-time)

Taught FORTRAN, BASIC, Pascal, and Introduction to Computing ▪ Developed two new courses (Pascal, C) ▪ Appointed department chair after one year (serving Summer semester).

*Adjunct Instructor*, Department of Computer Science, 1983, 1984 – 1988

Developed and taught courses in intermediate C programming and relational databases.

### **University of Arizona, Tucson, AZ**

*Adjunct Professor*, Department of Computer Science, 1982

Taught Advanced Scientific Programming.

### **California State University, Long Beach, CA**

*Adjunct Professor*, Department of Mathematics, 1980 – 1981

Taught Ordinary Differential Equations and Finite Mathematics.

### **Rancho Santiago College, Santa Ana, CA**

*Adjunct Instructor*, Department of Mathematics, 1980

Taught Numerical Methods.

### **Golden West College, Huntington Beach, CA**

*Adjunct Instructor*, Department of Mathematics and Computer Science, 1978 – 1980

Taught Beginning Algebra and APL Programming.

### **Brigham Young University, Provo UT**

*Instructor*, Department of Mathematics, 1977 – 1978 (full-time)

Taught College Algebra, Trigonometry and Linear Programming.

## **Industry Experience**

**2000 - 2001: Senior Software Engineer, [Novell, Inc.](#), Provo, UT**

As a member of the Custom Development Group, performed all phases of software engineering for small projects to serve Novell's clients ▪ Projects involved Java, C++, and Internet technologies

**1998 - 2000: Senior Technical Advisor, [Ingenix, Inc.](#), Salt Lake City, UT**

Provided mentoring and systems architecture consulting to projects using C++ and Oracle for the medical billing industry ▪ Taught internal courses on C++ and software engineering ▪ Developed a tool to automate the use of exceptions and the use of tables as objects in Pro-C++ ▪ Developed a framework for automated unit testing of C, C++ and Java programs

**1997 - present: President, [Fresh Sources, Inc.](#)**

President of a company offering training, mentoring, and custom programming services in object-oriented technology, specializing in C/C++, Java, Python, and Design Patterns ▪ Developed a multi-media training course in C programming (“Thinking in C”) distributed by [Mindview, Inc.](#)

**1990 - 1997: Consultant, [The Church of Jesus Christ of Latter-day Saints](#), Salt Lake City UT**

Development manager for the Church's first reusable, software infrastructure components project in a Windows client/server environment, using the Fusion object-oriented methodology, and Microsoft Visual C++ ▪ Conceived, designed, and implemented reusable, object-oriented frameworks for persistence and security for layered, object-oriented data models with relational databases ▪ Developed and championed software systems architecture ▪ Chairman of an in-house C Language Support Committee, which provided technical support, training, standards and direction for the use of C and C++ in software development ▪ Technical lead for a project that successfully automated the Church mission offices worldwide ▪ Designed and implemented a reporting subsystem in Oracle/ProC under HP-UX on-location for the Brazil Area Office

**1984 - 1990: Member of the Technical Staff, [Hughes Aircraft Company](#) (Raytheon), Tucson AZ**

Developed software tools for data reduction and document preparation in C ▪ Developed a software defect tracking system in a relational DBMS (INGRES on VAX/VMS) ▪ Directed software configuration management for the TOW Missile Program ▪ Participated in the development of a rule-based missile diagnostic system in a Symbolics Common LISP environment ▪ Developed a C-language library for PC network communications based on NetBios ▪ VMS system manager for two MicroVAX systems networked with PCs via TCP/IP ▪ Managed a PC open shop for employee users ▪ Gave on-site training in C, relational databases, numerical methods and object-oriented programming

**1981 - 1983: Research Assistant, [University of Arizona](#), Tucson AZ**

Performed scientific programming in FORTRAN and Ratfor under UNIX while a full-time Ph.D. candidate in Applied Mathematics (funded by a NASA Fellowship - finished all but dissertation with 4.0 gpa) ▪ Solved integral equations for water-flow research ▪ Wrote tools for automatic stability analysis of matrix algorithms

**1978 - 1981: Member of the Technical Staff, [Logicon, Inc. \(General Dynamics\)](#), San Pedro CA**

Developed software tools in FORTRAN in an IBM OS/MVS environment for the Space Defense Systems project ▪ Participated in the development of a software design/analysis tool written for IBM Corporation ▪ Developed software for weapons allocation and damage analysis studies ▪ Developed in-house training materials on APL and IBM's Structured Programming Facility software development environment

## ***Related Experience***

- Columnist and Technical Editor for *Better Software Magazine*, 2007 – 2008
- Software consulting for Homeland Security with Mission Support, Inc. (2006-2008)
- Developed and delivered Python training to software testers at Symantec, Inc. (2005, 2007)
- Editor at *The C/C++ Users Journal*, 1992-2003; **Senior Editor** 2001-2003
- Taught C++ and Java programming at various corporate sites for various training companies (Technology Exchange, Software Architects), 1993 – 1998
- Represented the Digital Equipment Corporation Users Society (DECUS), the Church of Jesus Christ of Latter-day Saints, and Utah Valley State College on the *ANSI/ISO Standards Committee* for the C++ Programming Language ▪ Designed part of the standard C++ library: **std::bitset** (1991-2001) ▪ Contributor to the C++ Boost Library: **Boost::dynamic\_bitset**
- Chair of the Industry Conference, *C/C++ Solutions*, R&D Publications, Kansas City, January 1995
- Developed and taught on-site courses in Numerical Methods, C Programming, Object-oriented Programming, and Relational Database Theory and Practice for Hughes Aircraft, 1987-1990

# Publications

## Books

1. Allison, C. (2021), *Foundations of Computing: An Accessible Introduction to Formal Languages*, Fresh Sources, Inc (ISBN 978-0-578-94417-3).
2. Eckel, B. & Allison, C. (2004), *Thinking in C++, Volume 2: Practical Programming*, Prentice-Hall, ISBN 0130353132 (Polish Translation, 2004; Chinese Translation 2005; Czech Translation, 2006).
3. Allison, C. (1998), *C & C++ Code Capsules*, Prentice-Hall, ISBN 0135917859. (Chinese translation 2002).

## Refereed Academic Papers

1. Hodgson, G., Harrison, N., Rudolph, G., Allison, C., Visualization of Requirements to Software Architecture Component Mappings, *Intermountain Engineering, Technology, and Computing Conference (i-ETC)*, October 2020.
2. Allison, C., Olson, K., On Countability, Enumeration, and How To Think Like a Computer Scientist, *The Journal of Computing Sciences in Colleges*, Volume 28, 1, October 2013.
3. Allison, C., Liddle, N., OOP: The Rest of the Story, *The Journal of Computing Sciences in Colleges*, Volume 26, 1, October 2011.
4. Allison, C., D: A Programming Language for Our Time, *The Journal of Computing Sciences in Colleges*, Volume 25, 1, October 2010.
5. Allison, C., Harrison, N., Teaching Design Patterns: A Matter of Principle, *The Journal of Computing Sciences in Colleges*, Volume 22, 1, October 2007.
6. Allison, C., The Untapped Power of Generic Algorithms, *The Journal of Computing Sciences in Colleges*, Volume 22, 1, October 2007.
7. Allison, C., Practical Computation Theory, *The Journal of Computing Sciences in Colleges*, Volume 22, 1, October 2007.
8. Allison, C., The Simplest Automated Unit Test Tool That Could Possibly Work, *The Journal of Computing Sciences in Colleges*, Volume 22, 1, October 2007.
9. Allison, C., Where Did All My Decimals Go?, *The Journal of Computing Sciences in Colleges*, Volume 20, 1, October 2005. Reprinted in February 2006.
10. Allison, C., On Determining Functions of Matrices, *Pi Mu Epsilon Journal*, Fall 1978.

## Peer-Reviewed Professional Articles

1. Allison, C., (2010) Floating-Point Numbers Aren't Real, invited contribution appearing in *97 Things Every Programmer Should Know*, Kevlin Henney, Editor, O'Reilly, 2010, ISBN 978-0-596-80948-5.
2. Allison, C. (2000 Sep) The Simplest Automated Unit Test Framework That Could Possibly Work, *C/C++ Users Journal*, 18, No. 9.
3. Cockburn, A., Allison, C. (1998 Jun) Object-oriented Analysis & Design, Part 2, *C/C++ Users Journal*, 1, No. 6.
4. Cockburn, A., Allison, C. (1998 May) Object-oriented Analysis & Design, Part 1, *C/C++ Users Journal*, 1, Nos. 5.
5. Allison, C. (1996 May) Object Persistence via Relational Databases, *C/C++ Users Journal*, 15, No. 5. (Reprinted as a *Dr. Dobbs* select publication in 1997).

## Other Professional Articles

1. Allison, C., (2008 May) A "D" in Programming, Part 2, *Better Software Magazine*, 10(4).
2. Allison, C., (2008 Apr) A "D" in Programming, Part 1, *Better Software Magazine*, 10(3).

3. Allison, C., (2008 Mar) Programming on the Other Side of Complexity, *Better Software Magazine*, 10(2).
4. Allison, C., (2008 Jan/Feb) Designing Resuable Software, *Better Software Magazine*, 10(1).
5. Allison, C., (2007 Dec) Tools for Our Times, *Better Software Magazine*, 9(12).
6. Allison, C. (2007 Nov) Buddy, Can You Paradigm?, *Better Software Magazine*, 9(11).
7. Allison, C. (2007 Sep) Neglected Algorithms: Making Reuse a Reality with STL, *Better Software Magazine*, 9(9).
8. Allison, C. (2007 Aug) Piles of Sand Redux, *Better Software Magazine*, 9(8).
9. Allison, C. (2007 Jun) Piles of Sand, *Better Software Magazine*, 9(6).
10. Allison, C. (2007 Apr) The Roof is Going to Go, *Better Software Magazine*, 9(4).
11. Allison, C. (2007 Feb) Principle-Centered Software Development, *Better Software Magazine*, 9(2).
12. Allison, C. (2001 Aug) Reflection, Java Supplement to the *C/C++ Users Journal*, 19, No. 8.
13. Allison, C. (2001 Jun) Threads 101, Java Supplement to the *C/C++ Users Journal*, 19, No. 6.
14. Allison, C. (2001 Apr) Understanding Java Exceptions, Java Supplement to the *C/C++ Users Journal*, 19, No. 4.
15. Allison, C. (2001 Feb) File Processing, Java Supplement to the *C/C++ Users Journal*, 19, No. 2.
16. Allison, C. (2000 Nov) Basic Stream I/O, *C/C++ Users Journal*, 18, No. 11.
17. Allison, C. (2000 Sep) Collections and Algorithms, *C/C++ Users Journal*, 18, No. 9.
18. Allison, C. (2000 Jul) Locales and Formatted I/O, *C/C++ Users Journal*, 18, No. 7.
19. Allison, C. (2000 Jun) Strings, *C/C++ Users Journal*, 18, No. 6.
20. Allison, C. (2000 Mar) Arrays, *C/C++ Users Journal*, 18, No. 3.
21. Allison, C. (2000 Jan) Interfaces and Inner Classes, *C/C++ Users Journal*, 18, No. 1.
22. Allison, C. (1999 Nov) Object-oriented Programming in Java, *C/C++ Users Journal*, 17, No. 11.
23. Allison, C. (1999 Sep) Packaging Your Objects, *C/C++ Users Journal*, 17, No. 9.
24. Allison, C. (1999 Jul) Thinking in Objects, *C/C++ Users Journal*, 17, No. 7.
25. Allison, C. (1999 May) Control Flow: The Bad, The Good, The Exceptional, *C/C++ Users Journal*, 17, No. 5.
26. Allison, C. (1999 Mar) Using Primitive Types and Wrappers, *C/C++ Users Journal*, 17, No. 3.
27. Allison, C. (1999 Jan) Jumping into Java, *C/C++ Users Journal*, 17, No. 1.
28. Allison, C. (1998 Dec) What's New in Standard C++?, *C/C++ Users Journal*, 16, No. 12.
29. Allison, C. (1997 Dec) Error Handling with C++ Exceptions, Part 2, *C/C++ Users Journal*, 15, No. 12.
30. Allison, C. (1997 Nov) Error Handling with C++ Exceptions, Part 1, *C/C++ Users Journal*, 15, No. 11.
31. Allison, C. (1996 Oct) C++: The Making of a Standard - Journey's End (An Interview with Bjarne Stroustrup), *C/C++ Users Journal*, 15, No. 10.
32. Allison, C. (1995 May) Data Abstraction, *C/C++ Users Journal*, 14, No. 5.
33. Allison, C. (1995 Apr) A Better C, *C/C++ Users Journal*, 14, No. 4.
34. Allison, C. (1995 Mar) The Standard C Library, Part 3, *C/C++ Users Journal*, 14, No. 3.
35. Allison, C. (1995 Feb) The Standard C Library, Part 2, *C/C++ Users Journal*, 14, No. 2.
36. Allison, C. (1995 Jan) The Standard C Library, Part 1, *C/C++ Users Journal*, 14, No. 1.
37. Allison, C. (1994 Dec) The Standard C++ Library, *C/C++ Users Journal*, 13, No. 12.
38. Allison, C. (1994 Nov) Dynamic Memory Management, Part 2, *C/C++ Users Journal*, 13, No. 11.
39. Allison, C. (1994 Oct) Dynamic Memory Management, Part 1, *C/C++ Users Journal*, 13, No. 10.
40. Allison, C. (1994 Sep) Conversions and Casts, *C/C++ Users Journal*, 13, No. 9.
41. Allison, C. (1994 Jul) C++ Exceptions, *C/C++ Users Journal*, 13, No. 7.
42. Allison, C. (1994 Jun) Control Structures, *C/C++ Users Journal*, 13, No. 6.
43. Allison, C. (1994 May) Visibility in C++, *C/C++ Users Journal*, 13, No. 5.
44. Allison, C. (1994 Apr) Visibility in C, *C/C++ Users Journal*, 13, No. 4.
45. Allison, C. (1994 Mar) The Preprocessor, *C/C++ Users Journal*, 13, No. 3.
46. Allison, C. (1994 Feb) Variable-length Argument Lists, *C/C++ Users Journal*, 13, No. 2.
47. Allison, C. (1994 Jan) Bit Handling in C++, Part 2, *C/C++ Users Journal*, 13, No. 1.
48. Allison, C. (1993 Dec) Bit Handling in C++, Part 1, *C/C++ Users Journal*, 12, No. 12.
49. Allison, C. (1993 Nov) Bit Handling in C, *C/C++ Users Journal*, 12, No. 11.

50. Allison, C. (1993 Oct) Pointers, Part 3, *C/C++ Users Journal*, 12, No. 10.
51. Allison, C. (1993 Sep) Pointers, Part 2, *C/C++ Users Journal*, 12, No. 9.
52. Allison, C. (1993 Aug) Pointers, Part 1, *C/C++ Users Journal*, 12, No. 8.
53. Allison, C. (1993 Jul) C++ Streams, *C/C++ Users Journal*, 12, No. 7.
54. Allison, C. (1993 Jun) File Processing, Part 2, *C/C++ Users Journal*, 12, No. 6.
55. Allison, C. (1993 May) File Processing, Part 1, *C/C++ Users Journal*, 12, No. 5.
56. Allison, C. (1993 Apr) Sorting with qsort, *C/C++ Users Journal*, 12, No. 4.
57. Allison, C. (1993 Mar) A C++ Date Class, Part 2, *C/C++ Users Journal*, 12, No. 3.
58. Allison, C. (1993 Feb) A C++ Date Class, Part 1, *C/C++ Users Journal*, 12, No. 2.
59. Allison, C. (1993 Jan) Time and Date Processing in C, *C/C++ Users Journal*, 12, No. 1.
60. Allison, C. (1992 Dec) Text Processing, Part 3, *C/C++ Users Journal*, 11, No. 12.
61. Allison, C. (1992 Nov) Text Processing, Part 2, *C/C++ Users Journal*, 11, No. 11.
62. Allison, C. (1992 Oct) Text Processing, Part 1, *C/C++ Users Journal*, 11, No. 10.

## **Editorials**

1. Allison, C. (August 2012) Not Your Father's C++, *Techwell.com*.
2. Allison, C., (October 2012) Google's Go – A Scalable Programming Language, *Techwell.com*.
3. Allison, C. (2008 Oct) A Gram of Prevention, *Better Software Magazine*, 10(7).
4. Allison, C. (2008 Jul/Aug) Software: Use At Your Own Risk, *Better Software Magazine*, 10(5).
5. Allison, C. (2005 Aug) As Simple As Possible?, *The C++ Source*
6. Allison, C. (2004 Sep) Your C++ Wish List, *The C++ Source*
7. Allison, C. (2004 Jun) C++ Reloaded, *The C++ Source*
8. Allison, C. (2003 Nov) The Incremental Developer, *C/C++ Users Journal*, 21, No. 11.
9. Allison, C. (2003 Oct) The Structure of Software Development Process Evolution, *C/C++ Users Journal*, 21, No. 10.
10. Allison, C. (2003 Sep) Human Activities, *C/C++ Users Journal*, 21, No. 9.
11. Allison, C. (2003 Aug) Programmer Dreams and Stranger Things, *C/C++ Users Journal*, 21, No. 8.
12. Allison, C. (2003 Jul) A Matter of Trust, *C/C++ Users Journal*, 21, No. 7.
13. Allison, C. (2003 Jun) Do Not Duplicate, *C/C++ Users Journal*, 21, No. 6.
14. Allison, C. (2003 May) Got Quality?, *C/C++ Users Journal*, 21, No. 5.
15. Allison, C. (2003 Apr) Stupid Tricks, *C/C++ Users Journal*, 21, No. 4.
16. Allison, C. (2003 Mar) The Proactive Programmer, *C/C++ Users Journal*, 21, No. 3.
17. Allison, C. (2003 Feb) The C++ Experience, *C/C++ Users Journal*, 21, No. 2.
18. Allison, C. (2003 Jan) The Wit to Win, *C/C++ Users Journal*, 21, No. 1.
19. Allison, C. (2002 Dec) Roots II, *C/C++ Users Journal*, 20, No. 12.
20. Allison, C. (2002 Nov) Part of the Landscape, *C/C++ Users Journal*, 20, No. 11.
21. Allison, C. (2002 Oct) Euclid Alone, *C/C++ Users Journal*, 20, No. 10.
22. Allison, C. (2002 Sep) The Best of Both?, *C/C++ Users Journal*, 20, No. 9.
23. Allison, C. (2002 Aug) New Cheese, *C/C++ Users Journal*, 20, No. 8.
24. Allison, C. (2002 Jul) O Brother, Where Art Thou?, *C/C++ Users Journal*, 20, No. 7.
25. Allison, C. (2002 Jun) Gear-head Humor and Other Important Items, *C/C++ Users Journal*, 20, No. 6.
26. Allison, C. (2002 May) Is Portability Still Important?, *C/C++ Users Journal*, 20, No. 5.
27. Allison, C. (2002 Apr) Good Stuff, *C/C++ Users Journal*, 20, No. 4.
28. Allison, C. (2002 Mar) SMOP, *C/C++ Users Journal*, 20, No. 3.
29. Allison, C. (2002 Feb) In-betweeners, *C/C++ Users Journal*, 20, No. 2.
30. Allison, C. (2002 Jan) A Simple Collection of Bits, *C/C++ Users Journal*, 20, No. 1.
31. Allison, C. (2001 Dec) What Every Programmer Should Know, *C/C++ Users Journal*, 19, No. 12.
32. Allison, C. (2001 Nov) We Have a Winner, *C/C++ Users Journal*, 19, No. 11.
33. Allison, C. (2001 Oct) Do Your Homework, *C/C++ Users Journal*, 19, No. 10.
34. Allison, C. (2001 Sep) This Sentence is False, *C/C++ Users Journal*, 19, No. 9.
35. Allison, C. (2001 Aug) The Name Game, *C/C++ Users Journal*, 19, No. 8.

36. Allison, C. (2001 Jul) Something Cool, *C/C++ Users Journal*, 19, No. 7.

## Citations

1. 155 citations of my scholarly work documented in various other publications (verify on *scholar.google.com*; search for “Chuck Allison”)
2. Two citations of my scholarly work in U.S. Patents:
  - a. US Patent 7644367 (“User Interface Automation Framework Classes and Interfaces”)
  - b. US Patent 6356957 (“Method for Emulating Native Object-oriented Foundation Classes on a Target Object-oriented Programming System using a Template Library”)
3. My software is used in many McAfee software products (see acknowledgement page in all user manuals; also available by online search, e.g. <https://docs.trellix.com/bundle/security-lotus-domino-v7-5-0-release-notes/resource/PD21876.pdf>)

## Lectures

### Keynote Lectures

1. “Confessions of a C++ Wonk”, *DConf2015*, May 2015.
2. “Engaged Engineering and Computer Science in Higher Education”, *American Society for Engineering Education* (ASEE), Rocky Mountain Chapter, April 2009.
3. “Tools for Our Times: Tales of Power, Intrigue and Serendipity in Computing”, Keynote Address at the Intermountain conference of the *Consortium for Computing Sciences in Colleges* (CCSC), October 2004.
4. “Practical Excellence: A Perspective on Code Quality,” Keynote Address, *Association of C/C++ Users* (Oxford, UK), April 2004.

### Invited Lectures

1. “A Project-Based Learning Approach to Teaching C++”, *CppCon*, September 2015.
2. “A Real D In Programming: Lessons Learned from Eight Years of Teaching D at the University”, *DConf* 2014, May 2014.
3. “Going Back to The Well: The Principles of Software Design”, *Better Software Conference*, June 2011.
4. “Coding For Keeps: How to Write a Function”, *Better Software Conference*, June 2011.
5. “The Design Principles Behind Design Patterns”, *Better Software Conference*, June 2008.
6. “Functional Programming Makes a Comeback”, *Better Software Conference*, June 2008.
7. “Fundamentals of Floating-point Arithmetic”, *In-Service Workshop for Teachers of Science and Engineering*, Utah Valley State College, June 2008.
8. “Totally Awesome Computing: Python as a General-Purpose Object-Oriented Programming Language”, *ACM/OOPSLA*, 2006, 2007.
9. “Understanding C++ Templates”, *Software Development Conference*, March 2005 – 2008, October 2006, 2007.
10. “Where Did All My Decimals Go?”, *Software Development Conference*, March 2006.
11. “Got Quality?”, *Software Development Conference*, March 2005, 2006.
12. “Understanding C++ Templates,” *Association of C/C++ Users* (Oxford, UK), April 2004.
13. “Automated Testing”, *Provo Linux Users Group*, Orem, UT, January 2003.
14. “The Forgotten Containers”, *Software Development Conference*, 2002.
15. “Understanding C++ Exception Handling”, *Software Development Conference*, 1997 - 2003.
16. “Multithreading in Java”, *Software Development Conference*, 2001-2003.
17. “The Simplest Automated Unit Test Framework That Could Possibly Work”, *Software Testing, and Review Conference (STAR)*, 2001; also *Software Development Conference*, 2001.
18. “The Java 2 Collections”, *Software Development Conference*, 1999-2001.
19. “Object Persistence with Relational Databases”, *Software Development Conference*, 1997.
20. “Practical STL”, Workshop for *Software Development Conference*, 1997.

21. “Leveraging the Standard C++ Library”, *Software Development Conference*, 1994-1999.
22. Gave several invited lectures on C++ programming at symposia of the *Digital Equipment Computer Users Society (DECUS)*, 1991-1993.

## Professional Development

- Attended *SciPy*, Austin, Texas, July 2015; July 2018.
- Attended *PyCon*, the Python Community Annual Conference, Washington, DC, March, 2005; May 2018
- Attended *CppCon*, Bellevue, Washington, September 2015, 2016, 2017.
- *Introduction to Swift Programming* by University of Toronto on Coursera. Certificate earned December 2015
- Attended the National Computational Science Institute’s (*NCSI*) *Computational Thinking Workshop*, Patchogue, New York, June 2014, and Orem, Utah, June 2015.
- Attended the *D Conference (DConf)*, 2007, 2013, 2014, 2015
- Attended Think Forward’s *Project-Based Learning Workshop*, Austin, Texas, June 2014.
- Attended regional conferences of *The Consortium for Computing Sciences in Colleges (CCSC)*, 2004, 2005, 2007, 2010, 2011, 2013
- Attended *C++ Now* Conference, Aspen, CO, May 2012
- Attended the annual Symposium of *SIGCSE*, the Special Interest Group for Computer Science Education, Milwaukee, WI, March 2010; Reno, NV, February 2003
- Attended *The Teaching Professor Conference*, Washington, DC, June 2009
- Attended the *International Conference on Functional Programming*, Victoria, BC, September 2008
- Attended various educational talks and tutorials at *Software Development Conference* twice yearly, 1994 – 2008 (various US locations)
- Attended *C++ Connections*, Las Vegas, NV, November 2005
- Attended *Java One*, San Francisco, CA, June 2004
- Attended the *Agile Development Conference*, Salt Lake City, UT, June 2003
- Co-Convener of “Summit on Code Quality”, Portland, OR, January 2003
- Attended Mindview’s seminar *Advanced Java Programming*, Crested Butte, CO, July 1998
- Attended *DECUS Symposia* semi-annually, 1991-1993 (Digital Equipment Corporation Users Society, various US locations)