

Thomas VanDrunen

Curriculum vitæ

Mathematics and Computer Science
Wheaton College
501 College Ave, Wheaton, IL 60187
☎ +1 630 752 5692 (office)
📞 +1 630 639 2255 (mobile)
✉ Thomas.VanDrunen@wheaton.edu
🌐 <http://cs.wheaton.edu/~tvandrun>
Updated July 26, 2017

Personal information

Citizenship USA

Education

- 2004 **PhD**, *Computer Science*, Purdue University, West Lafayette, IN.
Dissertation: *Partial Redundancy Elimination for Global Value Numbering*.
Academic advisor: Antony Hosking.
- 2000 **MSc**, *Computer Science*, Purdue University, West Lafayette, IN.
- 1998 **BSc**, *Mathematics and Computer Science*, Calvin College, Grand Rapids, MI.

Teaching interests

Data structures and algorithms; computational linguistics; discrete mathematics supplemented with functional programming; programming languages and compilers; introductory programming; computer science for non-majors, especially in a liberal arts context.

Research interests

Natural language processing; programming languages: foundations, design, implementation, compiler construction tools, optimization, and comparison and hybridization of paradigms; object-oriented software engineering: design patterns, development tools.

Professional experience

- Since 2011 **Associate Professor with tenure** *Department of Mathematics and Computer Science*, Wheaton College, Wheaton, IL.
- 2004–2011 **Assistant Professor** *Department of Mathematics and Computer Science*, Wheaton College, Wheaton, IL.
- 2001–2004 **Graduate Research Assistant**, *Department of Computer Science*, Purdue University, West Lafayette, IN.
- 2003 **Graduate Teaching Assistant**, for CS 502 Compilers, *Department of Computer Science*, Purdue University, West Lafayette, IN.

- 2001 **Instructor**, for CS 180 (Introduction to Programming) *Department of Computer Science*, Purdue University, West Lafayette, IN.
- 1998 **Undergraduate Fellow**, *Department of Mathematics and Computer Science*, Argonne National Laboratory, Argonne, IL.

Publications

- Magazine article – 2017 “Functional Programming as a Discrete Mathematics Topic,” in *ACM Inroads* 8(2):51–58, May 2017.
- Book – 2012 *Discrete Mathematics and Functional Programming*, Franklin, Beedle and Associates. August 2012.
- Conference paper – 2011 “The Case for Teaching Functional Programming in Discrete Math,” in the proceedings of the Educators’ Symposium at Systems, Programming, Languages and Applications: Software for Humanity (SPLASH), October 2011.
- Journal article – 2010 “A Christian Analysis of Gabriel’s ‘Mob Software,’ ” *The Journal of the ACMS*, 2010-2011 issue.
- Journal article – 2008 “Wake up and smell the coffee: Evaluation methodology for the 21st century,” with S. Blackburn et al, in *Communications of the ACM* 51(8):83–89.
- Conference paper – 2006 “The DaCapo benchmarks: Java benchmarking development and analysis,” with S. Blackburn et al., in the proceedings of Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), October 2006.
- Conference paper – 2006 “Java Interfaces in CS 1 Textbooks,” in the proceedings of the Educators’ Symposium at Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), October 2006.
- Book chapter – 2005 “How is God’s Creativity Manifested in Computer Science?” in *Not Just Science*, edited by Dorothy Chappell and David Cook.
- Journal article – 2004 “Anticipation-based partial redundancy elimination for static single assignment form,” with A. L. Hosking. *Software—Practice and Experience*, 34(15).
- Conference paper – 2004 “Value-Based Partial Redundancy Elimination,” with A. L. Hosking, in the proceedings of the Thirteenth International Conference on Compiler Construction (CC), March 2004.
- Workshop paper – 2004 “Visitor-Oriented Programming,” with J. Palsberg, in the proceedings of the Eleventh Workshop on Foundations of Object-Oriented Languages (FOOL), Jan 2004.

Teaching

Courses taught at Wheaton College

- Programming **CSCI 235: Programming I: Problem-Solving.** Innovations include the development of a laboratory section, the reordering of material on polymorphism, and the replacement of an earlier course with this, the first in a two-semester sequence. (Fall 2009, Fall 2008, Spring 2007, Fall 2006; similar course Spring 2006, Fall 2005, Spring 2005, Fall 2004; similar course at Purdue University, Summer 2001.)

Theory	CSCI/MATH 243: Discrete Mathematics and Functional Programming. Developed course as a foundational component in a thorough revision of the computer science curriculum. Innovations include composing a course text and pioneering the teaching of discrete mathematics and functional programming (using the ML programming language) in a single course. (Spring 2017, Fall 2016, Fall 2015, Fall 2014, Fall 2013, Fall 2012, 2011, Fall 2010, Fall 2009, Fall 2008, Fall 2007, Fall 2006, Fall 2005, Spring 2005.)
Programming	CSCI 245: Programming II: Object-Oriented Design. New course, designed and taught for the first time as a foundational component in a thorough revision of the computer science curriculum. Innovations include comparing theoretical analysis of algorithms with instrumentation of implementations, design patterns, and an introduction to systems programming in C. (Spring 2016, Fall 2015, Spring 2014, Fall 2013, Fall 2012, Spring 2012, Fall 2011, Spring 2011, Fall 2010, Spring 2010, Spring 2009, Spring 2008, Fall 2007, Spring 2007.)
Software engineering	CSCI 335: Software Development. New course, designed and taught for the first time as a foundational component in a thorough revision of the computer science curriculum. Innovations include extensive coverage of design patterns, a semester-long group project, and student presentations of textbook material. (Spring 2014, Spring 2012, Spring 2011, Spring 2010, Spring 2009, Fall 2007.)
Programming & theory	CSCI 345: Data Structures and Algorithms. Innovations include designing a new comprehensive series of projects, emphasizing the interplay between theory and experimentation when studying correctness and efficiency, and adding the topic of dynamic programming. (Spring 2017, Spring 2016, Spring 2015.)
Programming languages	CSCI 365: Programming Language Concepts. Innovations include an emphasis on theory and foundations and a highly-developed sequence of implementation projects. (Spring 2016, Spring 2014, Spring 2012, Spring 2010, Spring 2008, Spring 2006.)
Language technology	CSCI 384: Computational Linguistics. Taught and designed as a new course. Topics include statistical language modeling and applications across the liberal arts. (Fall 2015, Fall 2013.)
Programming	CSCI 394: Advanced Programming. Special topics course, taught with this topic for the first time. Topics in this course were later incorporated into other courses in the curriculum, especially CSCI 335. (Spring 2005.)
Theory	CSCI 394: Advanced Discrete Mathematics and Functional Programming Seminar. Special topics course, taught with this topic for the first time. Topics include graph theory, lattices, group theory, RSA encryption, automata theory, and advanced functional programming techniques. (Spring 2011, Spring 2008.)
Theory	CSCI 445: Analysis of Algorithms. Innovations include empirical projects to compare with results of theoretical findings and a module on the theory of computation. (Fall 2016, Fall 2014, Fall 2012, Fall 2010, Fall 2008, Fall 2006, Fall 2004.)
Integrative issues	CSCI 494: Social and Ethical Issues in Computing. (Spring 2017, Spring 2009.)

Institutional service

Committee and administrative work at Wheaton College

- 2013– Serving as Computer Science **Program Coordinator**.
- 2007–2015 Served on the **Faculty Financial Affairs Committee** (chair during the 2008–2009 academic year).
- 2008–2009 Served on **Faculty Council**.
- 2005–2007 Served on the **Hearing Panel for Student Conduct** (chair during the 2006–2007 academic year).

Other professional activities

- Award Received Wheaton College senior faculty teaching achievement award (2017).
- Award Jointly received most influential paper at OOPSLA 2006 (award presented at OOPSLA 2016).
- Peer institutional service Served as external reviewer for the computer science program at Covenant College (Spring 2015).
- Peer institutional service Served as external member for computer science curricular revision advisory committee at Trinity Christian College (Summer and Fall 2013)
- Continuing education Attended courses on natural language processing at the Language Technology Institute at Carnegie Mellon University (sabbatical, Spring 2013)
- Continuing education Attended seminar on teaching discrete mathematics at Messiah College (Summer 2006)
- Award Received Computer science graduate teaching assistant award (2003)
- Continuing education Attended International School on Foundations of Security Analysis and Design (FOSAD) (Fall 2000).