Marcelo d'Amorim

Associate Professor, Computer Science Department
North Carolina State University

Address
890 Oval Drive 3320 Engineering Building II Raleigh, NC 27606

Email ..... mdamori@ncsu.edu
Personal Page ..... https://damorim.github.io
Skype id ..... mdamorim
Work Phone ..... +1 (919) 515-3644

Education

University of Illinois at Urbana-Champaign ..... Champaign, IL, US
2007 Ph.D. in Computer Science
Dissertation title: "Efficient Explicit-State Model Checking for Programs with Dynamically Allocated Data"
Advisor: Darko Marinov

Federal University of Pernambuco ..... Recife, PE, Brazil
2001 M.S. in Computer Science
1997 B.S. in Computer Science

Research Interests My research interests are in the areas of **Software Engineering and Programming Languages**, with a focus on improving software reliability through program analysis and systematic testing.

Experience

'22– **North Carolina State University** ..... Raleigh, NC, US
Associate Professor. Advising: 4PhD (1 from UFPE), Co-advising: 1PhD.

'09–'22 **Federal University of Pernambuco (UFPE)** ..... Recife, PE, Brazil
Associate Professor. Graduated: 1PhD+8MS, Co-advised: 4MS.

'15–'16 **Georgia Institute of Technology** ..... Atlanta, GA, US
Visiting Scholar at the Arktos group led by Alessandro Orso

'07–'08 **Federal University of Pernambuco** ..... Recife, PE, Brazil
Postdoctoral researcher at the SPG group led by Paulo Borba

Software https://github.com/ncsu-swat

Service https://damorim.github.io/service.html

Current PhD Students

'22- M. M. Abid Naziri, Faster and More Effective Detection of ADS Failures
'22- Ishrak Hayet, Practical Generation of Software Specifications
'22- Ummay Kulsum, Automated Repair of Software Vulnerabilities with LLMs
'21- Denini Silva (UFPE), Shaker: Practical Detection of Flaky Tests with Noise
'16- Sarah Elder (Co-advised), Aggregation Methods of Score Cards to Support Supply-Chain Security Analysis

Graduated Students

MS'23 Paulo Nunes, Detecting Failures in Autonomous Driving with XAI
Now: Software Architect at Stelantis.

MS'23 Beatriz Souza, Automated Detection of Code-Comment Inconsistencies
Now: PhD student at University of Stuttgart, Germany

MS'20 Igor Lima, Leveraging Diversity to Find Bugs in JavaScript Engines
Now: Software Developer at Sensedia, Brazil

MS'19 Luis Melo, Using Docker to Assist QA Forum Users
Now: Software Development Engineer at AWS, Canada

MS'18 Jeanderson Cândido, Test Suite Parallelization in Open-Source Projects
Now: PhD student at TU Delft, Netherlands

MS'16 Paulo Barros, Resolving Java Reflection and Android Intents
Now: Senior Software Engineer at Chronicled, Brazil

MS'15 Mateus Borges, qCORAL: Quantitative Constraint Solver for Complex Mathematical Constraints
Now: Senior Backend Engineer at Elinvar Gmbh, Germany

PhD'15 Sabrina Souto, Addressing High Dimensionality and Lack of Feature Models in Testing of Software Product Lines
Now: Assistant Professor at UEPB, Brazil

MS'12 Elton Alves, Improved Fault Localization with Dynamic Slicing and Change Impact Analysis
Now: Senior Software Engineer at Zartis, Spain

MS'12 João Paulo Oliveira, Rabbit - A Novel Approach to Find Data Races in Concurrent Programs
Co-advised with Fernando Castor, Now: CEO NoxBitcoin, Brazil

MS'10 Andrei Rimsa Alvares, Efficient Static Analysis to Find Tainted Variable Attacks
Co-advised with Fernando Pereira and Roberto Bigonha, Now: Assistant Professor at CEFET-MG, Brazil

MS'09 Mitsuo Takaki, Effective CSP solvers with Particle-Swarm Optimization and Genetic Algorithms
Co-advised with Ricardo Prudêncio, Now: Principal Software Engineer at Absolute Software, Canada

MS'08 Gláucia Peres, A Black-box Testing Technique for the Detection of Crashes Based on Automated Test Scenarios
Co-advised with Alexandre Mota, Now: Director of Engineering at FreshBooks, Canada

Funding

Overall, I raised (1) ∼$704K from 10 international, national, and state grants, (2) ∼$161K for student stipends, and (3) ∼$16K from university awards and gifts.

24-27, Co-PI, NSF, SLES: Context-Aware Safe (CAS) Lower-Limb Robotic Prosthesis. $1.50M.
24-27, PI, NSF, SHF: Small: E2R2, A Comprehensive Approach to Improve Simulation-based Testing of Autonomous Driving Systems. $559K.
23-25, PI, NSF, Collaborative Research: FMitF: Track II: Cross-Language Support for Runtime Verification (CNS-2026928). $50K.
20-24, PI, NSF, Collaborative Research: SaTC: TTP: Small: eSLIC: Enhanced Security Static Analysis for Detecting Insecure Configuration Scripts. $199.9K.

Conference Publications

(*) Name of students that I supervised appear underlined.

(**) Selected papers are labeled with an asterisk.

ASE'22 [1] Sofia Reis, Rui Abreu, **M. d'Amorim**, and Daniel Fortunato. Leveraging Practitioners' Feedback to Improve a Security Linter.In *IEEE/ACM Intl. Conference on Automated Software Engineering (ASE)*, 2022.

ICSE'22 [2] Facundo Molina, Nazareno Aguirre, and **M. d'Amorim**. Fuzzing Class Specifications. In *International Conference on Software Engineering (ICSE). (To Appear)* May 2022.

ROPES'22 [3] Brittany Reid, Markus Wagner, **M. d'Amorim**, and Christoph Treude. Software Engineering User Study Recruitment on Prolific: An Experience Report. In *International Workshop on Recruiting Participants for Empirical Software Engineering (ROPES). (To Appear)* May 2022.

ICSME'21 [4] Shouvick Mondal, Denini Silva, and **M. d'Amorim**. Soundy Automated Parallelization of Test Execution. In *IEEE International Conference on Software Maintenance and Evolution (ICSME)*. pages 309–319. Sept. 2021

*ICSE'21[5] Jordan Henkel, Denini Silva, Leopoldo Teixeira, **M. d'Amorim**, and Thomas Reps. Shipwright: A Human-in-the-Loop System for Dockerfile Repair. In *International Conference on Software Engineering (ICSE)*. pages 1148–1160. May 2021.

ICST'21 [6] Leopoldo Teixeira, Breno Miranda, Henrique Rebêlo, and **M. d'Amorim**. Demystifying the Challenges to Formally Specifying API Properties for Runtime Verification. In *IEEE International Conference on Software Testing, Verification and Validation (ICST)*. pages 82–93. April 2021.

ICSME'20 [7] Denini Silva, Leopoldo Teixeira and **M. d'Amorim**. Shake It! Detecting Flaky Tests Caused by Concurrency with Shaker. In *International Conference on Software Maintenance and Evolution (ICSME)*. pages 492–502, September 2020.

ICST'20a [8] Breno Miranda, Igor Lima, Owolabi Legunsen, and **M. d'Amorim**. Prioritizing Runtime Verification Violations. In *IEEE International Conference on Software Testing, Verification and Validation (ICST)*. pages 297-308, October 2020.

ICST'20b [9] Marcio A. Guimarães, Leo Fernandes, Márcio Ribeiro, **M. d'Amorim**, and Rohit Gheyi. Optimizing Mutation Testing by Discovering Dynamic Mutant Subsumption Relations. In *IEEE International Conference on Software Testing, Verification and Validation (ICST)*. pages 198–208, October 2020.

MSR'20 [10] Gustavo Pinto, Breno Miranda, Supun Dissanayake, **M. d'Amorim**, Christoph Treude, and Antonia Bertolino What is the Vocabulary of Flaky Tests? In *International Conference on Mining Software Repositories (MSR)*. pages 492–502, July 2020.

ICSE-NIER'20 [11] **M. d'Amorim**, Rui M. Abreu, and Carlos Mello. Visual Sketching: From Image Sketches to Code. In *International Conference on Software Engineering New Ideas and Emerging Results (ICSE NIER)*. pages 101–104, July 2020.

IJCAI'19 [12] Sofia Reis, Rui Abreu, and **M. d'Amorim**. A Study of Demystifying the Combination of Dynamic Slicing and Spectrum-based Fault Localization. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4760-4766, August 2019.

ICST'19 [13] Xiangyu Li, **M. d'Amorim**, and Alessandro Orso. Intent-Preserving Test Repair. In *IEEE International Conference on Software Testing, Verification and Validation (ICST)*, pages 217–227. April 2019.

ISSTA'18 [14] Mattia Fazzini, Martin Prammer, **M. d'Amorim**, and Alessandro Orso. Automatically Translating Bug Reports into Test Cases for Mobile Apps. In *International Symposium on Software Testing and Analysis (ISSTA)*, pages 141-152, July 2018.

*ICSE'18[15] Xiangyu Li, Shaowei Zhu, **M. d'Amorim**, and Alessandro Orso. Enlightened Debugging. In *International Conference on Software Engineering (ICSE)*, pages 82-92, May 2018.

ASE'17 [16] Jeanderson Candido, Luis Melo, **M. d'Amorim**. Test Suite Parallelization in Open-Source Projects: a Study on its Usage and Impact. In *IEEE/ACM Intl. Conference on Automated Software Engineering (ASE)*, pages 838-848, Nov. 2017.

ICSE'17 [17] Sabrina Souto, **M. d'Amorim**, Rohit Gheyi. Balancing Soundness and Efficiency for Practical Testing of Configurable Systems. In *International Conference on Software Engineering (ICSE)*, pages 632-642, May 2017.

ICST'17 [18] Alexandre Perez, Rui Abreu, **M. d'Amorim**. Prevalence of Single-Fault Fixes and its Impact on Fault Localization. In *IEEE International Conference on Software Testing, Verification and Validation (ICST)*, pages 12-22, March 2017.

HVC'16 [19] Xiangyu Li, **M. d'Amorim**, Alessandro Orso. Iterative User-Driven Fault Localization. In *Haifa Verification Conference (HVC)*, pages 82-98, November 2016.

ASE'15 [20] Paulo Barros, René Just, Suzanne Millstein, Paul Vines, Werner Dietl, **M. d'Amorim**, and Michael D. Ernst. Static Analysis of Implicit Control Flow: Resolving Java Reflection and Android Intents. In *IEEE/ACM Intl. Conference on Automated Software Engineering (ASE)*, pages 669-679, Nov. 2015.

ESEC-FSE'15 [21] Mateus Borges, Antonio Filieri, **M. d'Amorim**, and Corina S. Păsăreanu. Iterative Distribution-Aware Sampling for Probabilistic Software Analysis. In *European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE)*, pages 866-877, Sept. 2015.

SPLC'15 [22] Sabrina Souto, Divya Gopinath, **M. d'Amorim**, Darko Marinov, Sarfraz Khurshid and Don Batory. Faster Bug Detection for Software Product Lines with Incomplete Feature Models. In *International Systems and Software Product Line Conference (SPLC)*, pages 151–160, July 2015.

HVC'14 [23] Tianhai Liu, Mateus Borges, **M. d'Amorim**, and Mana Taghidiri. A Comparative Study of Incremental Constraint Solving Approaches in Symbolic Execution. *Haifa Verification Conference (HVC)*, pages 284-299, Nov. 2014.

SPIN'14 [24] Quoc-Sang Phan, Pasquale Malacaria, Corina S. Păsăreanu, and **M. d'Amorim**. Quantifying Information Leaks using Reliability Analysis. *International SPIN Symposium on Software Model Checking (SPIN)*, pages 105–108, July 2014.

*PLDI'14[25] Mateus Borges, Antonio Filieri, **M. d'Amorim**, Corina S. Păsăreanu, and Willem Visser. Compositional Solution Space Quantification for Probabilistic Software Analysis. *ACM/SIGPLAN Programming Language Design and Implementation (PLDI)*, pages 123-132, June 2014.

ASE'13 [26] José Carlos de Campos, Rui Abreu, Gordon Fraser, and **M. d'Amorim**. Entropy-based Test Generation for Improved Fault Localization. In *IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 257–267, November 2013.

ESEC-FSE'13 [27] Chang Hwan Peter Kim, Darko Marinov, Sarfraz Khurshid, Don Batory, Sabrina Souto, Paulo Barros, and **M. d'Amorim**. SPLat: Lightweight Dynamic Analysis for Reducing Combinatorics in Testing Configurable Systems. In *European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE)*, pages 257–267, August 2013.

ICST'12 [28] Mateus Borges, **M. d'Amorim**, Saswat Anand, David Bushnell, and Corina S. Păsăreanu. Symbolic Execution with Interval Solving and Meta-heuristic Search. In *IEEE International Conference on Software Testing, Verification, and Validation (ICST)*, pages 111-120. April 2012.

ASE'11 [29] Elton Alves, Milos Gligoric, Vilas Jagannath, and **M. d'Amorim**. Improved Lightweight Debugging with Dynamic Slicing and Change Data. In *IEEE/ACM International Symposium on Automated Software Engineering (ASE)*, pages 520-523, Nov. 2011.

NFM'11 [30] Matheus Souza, Mateus Borges, **M. d'Amorim**, and Corina S. Păsăreanu. CORAL: Solving Complex Constraints in Symbolic PathFinder. In *Proc. of the NASA Formal Methods Symposium (NFM)*, pages 359-374, April 2011.

CC'11 [31] Andrei Rimsa, **M. d'Amorim**, Fernando M. Q. Pereira. Efficient Tainted Flow Analysis. In *ETAPS Intl. Conference on Compiler Construction (CC)*, pages 124-143, March 2011.

CbSoft'10 [32] Andrei Rimsa, **M. d'Amorim**, Fernando M. Q. Pereira. Efficient Static Checker for Tainted Variable Attacks. In *Brazilian Symposium on Programming Languages (SBLP)*, September 2010.

NFM'09 [33] M. Takaki, D. Cavalcanti, R. Gheyi, J. Iyoda, **M. d'Amorim**, R. Prudencio. A Comparative Study of Randomized Constraint Solvers for Random-Symbolic Testing. In *NASA Formal Methods Symposium (NFM)*, , pages 56-65, April 2009.

ICST'09 [34] C. Bertolini, G. Peres, **M. d'Amorim**, A. Mota. An Empirical Evaluation of Automated Black Box Testing Techniques for Crashing GUIs. In *IEEE International Conference on Software Testing, Verification, and Validation (ICST)*, pages 21-30, April 2009.

ISSTA'07 [35] **M. d'Amorim**, S. Lauterburg, and D. Marinov. Delta Execution for Efficient State-Space Exploration of Object-Oriented Programs. In *ACM/SIGSOFT International Symposium on Software Testing and Analysis (ISSTA)*, pages 50-60. July 2007.

ASE'06 [36] **M. d'Amorim**, C. Pacheco, T. Xie, D. Marinov, and M. D. Ernst. An empirical comparison of automated generation and classification techniques for object-oriented unit testing. In *IEEE/ACM International Symposium on Automated Software Engineering (ASE)*, pages 59-68, September 2006.

ICFEM'06 [37] **M. d'Amorim**, A. Sobeih, and D. Marinov. Optimized execution of deterministic blocks in Java PathFinder. In *International Conference on Formal Engineering Methods (ICFEM)*, pages 549-567. November 2006.

CAV'05 [38] **M. d'Amorim** and G. Roşu. Efficient Monitoring of Omega-Languages. In *Intl. Conference on Computer Aided Verification (CAV)*. pages 364-378. July 2005.

SBLP'05 [39] **M. d'Amorim** and G. Roşu. An Equational Specification for the Scheme Language. In *Simpósio Brasileiro de Linguagens de Programação (SBLP)*. pages 229-242, June 2005.

ICFEM'04 [40] F. Chen, **M. d'Amorim**, and G. Roşu. A Formal Monitoring-based Framework for Software Development and Analysis. In *Proc. of the International Conference on Formal Engineering Methods (ICFEM)*. pages 357-372, November 2004.

Journal Publications

- [41] Silva, D., Miranda, B., Teixeira, L., d'Amorim, M., Using Noise to Detect Test Flakiness. *submitted*

- [42] Silva, D., Gruber, M., Gokhale, S., Arteca, E., Turcotte, A., d'Amorim, M., Lam, W., Winter, S., Bell, J., The Effects of Computational Resources on Flaky Tests, *submitted*

- [43] Naziri, M. M. A., Lambertenghi, S. C., Stocco, A., d'Amorim, M., Misbehaviour Forecasting for Focused Autonomous Driving Systems Testing. *submitted*

- [44] Hayet, I., d'Amorim, M., ChatAssert: LLM-based Test Oracle Generation with External Tools Assistance. *submitted*

TSE'23 [45] Torres, A., Costa, P., Amaral, L., Pastro, J., Rodrigo, B., Legunsen, O., d'Amorim, M., Bodden, E., Dias, E., Runtime Verification of Crypto APIs: An Empirical Study.

TSE'23 [46] Brittany Reid, Keila Barbosa, **M. d'Amorim**, Markus Wagner, and Christoph Treude. NCQ: Code Reuse Support for Node.js Developers. In *Transactions of Software Engineering*, 2023

STVR'22 [47] Lucas Cabral, Breno Miranda, Igor Lima, and **M. d'Amorim**. RVPrio: A Tool for Prioritizing Runtime Verification Violations. In *Software Testing, Verification and Reliability (STVR)*. 2022

TREL'22 [48] Lucas Alcantara, Guilherme Padilha, Rui Abreu, and **M. d'Amorim**. Syrius: Synthesis of Rules for Intrusion Detectors. In *IEEE Transactions on Software Reliability (TRel)*. March 2022.

IST'21 [49] Rohit Gheyi, Márcio Ribeiro, Beatriz Sousa, Marcio Guimarães, Leo Fernandes, **M. d'Amorim**, Vander Alves, Leopoldo Teixeira, and Baldoino Fonseca. Identifying Method-Level Mutation Subsumption Relations using Z3. In *Information and Software Technology (IST)*. November 2021.

JSS'20 [50] Igor Lima, Jefferson Silva, Breno Miranda, Gustavo Pinto, and **M. d'Amorim**. Exposing Bugs in JavaScript Engines through Test Transplantation and Differential Testing. In *Journal of Systems and Software (JSS)*. November 2020.

IST'20 [51] Igor Lima, Jeanderson Candido, and **M. d'Amorim**. Practical Detection of CMS Plugin Conflicts in Large Plugin Sets. In *Information and Software Technology (IST)*. February 2020.

TSE'20 [52] Luis Melo, Igor Wiese, and **M. d'Amorim**. Using Docker to Assist Q&A Forum Users. In *IEEE Transactions on Software Engineering (TSE)*. December 2019.

JSS'17 [53] Sabrina Souto and **M. d'Amorim**. Time-Space Efficient Regression Testing for Configurable Systems. In *Journal of Systems and Software (JSS)*, Volume 137, pages 733-746, 2018.

SCP'14 [54] Andrei Rimsa, **M. d'Amorim**, Fernando M. Q. Pereira, and Roberto S. Bigonha. Efficient Static Checker for Tainted Variable Attacks. *Science of Computer Programming (SCP)*. Volume 80, pages 91–105, Feb. 2014

SIMULATION'10 [55] A. Sobeih, **M. d'Amorim**, M. Viswanathan, D. Marinov, and J. Hou. Assertion checking in J-Sim simulation models of network protocols. In *Transactions of The Society for Modeling and Simulation International (Simulation)*. Volume 86, Number 11, 651-673, November 2010.

ISSE'10 [56] M. Takaki, D. Cavalcanti, R. Gheyi, J. Iyoda, **M. d'Amorim**, R. Prudencio. Randomized Constraint Solvers: A comparative study. In *Innovations in Systems and Software Engineering: a NASA journal (ISSE)*. Volume 6, Number 3, 243-253, September 2010.

TSE'08 [57] **M. d'Amorim**, S. Lauterburg and D. Marinov. Delta Execution for Efficient State-Space Exploration. In *IEEE Transactions on Software Engineering (TSE)*, Vol. 34, No. 5, pages 597-613, October 2008.

JUCS'05 [58] **M. d'Amorim** and G. Roşu. An Equational Specification for the Scheme Language. In *Journal of Universal Computer Science (JUCS)*, 11(7), pages 1327-1348, July 2005.

Tool Demos + Workshops

ASED'21[59] Marcello Cordeiro, Denini Silva, Leopoldo Teixeira, Breno Miranda, and **M. d'Amorim**. Shaker: A Tool for Detecting More Flaky Tests Faster. In *IEEE/ACM International Conference on Automated Software Engineering (ASE Tool Demonstrations)*. November 2021.

DocEng'21 [60] Daniela Costa, Carlos Mello, and **M. d'Amorim**. A Comparative Study on Methods and Tools for Handwritten Mathematical Expression Recognition. In *ACM Symposium on Document Engineering (DocEng)*. August 2021.

SafeThings'19 [61] Davino Mauro Junior, Luis Melo, Harvey Lu, **M. d'Amorim**, and Atul Prakash. A Study of Vulnerability Analysis of Popular Smart Devices Through Their Companion Apps. In *IEEE Workshop on the Internet of Safe Things (SafeThings)*, May 2019.

ICSED'08[62] T. Gvero, M. Gligoric, S.Lauterburg, **M. d'Amorim**, D. Marinov, S. Khurshid State Extensions for Java PathFinder. In *International Conference on Software Engineering (ICSE Demo)*. pages 863-866, May 2008.

HotDep'07 [63] Y. Zhou, D. Marinov, W. Sanders, C. Zilles, **M. d'Amorim**, S. Lauterburg, R. Lefever, J. Tucek Delta Execution for Software Reliability. In *Workshop on Hot Topics in System Dependability (HotDep)*, June 2007.

RV'05 [64] F. Chen, **M. d'Amorim** and G. Roşu. Checking and Correcting Behaviors of Java Programs at Runtime with Java-MOP. In *5th Workshop on Runtime Verification (RV)*, pages 3-20, July 2005.

WODA'05 [65] **M. d'Amorim** and K. Havelund. Event-Based Runtime Verification of Java Programs. In *ACM/SIGSOFT International Workshop on Dynamic Analysis (WODA)*, pages 15-21, July 2005.

WGP'02 [66] **M. d'Amorim**, C. Nogueira, G. Santos, A. Souza, and P. Borba. Integrating Code Generation and Refactoring. In *Proc. of the Workshop on Generative Programming* (ECOOP event), June 2002.

WLM-PSC'01 [67] **M. d'Amorim** and C. Ferraz. Designing Jini Distributed Services: A Framework to support the development of reliable component networks. In *Proc. of the Workshop on Language Mechanisms for Programming Software Components* (OOPSLA event), October 2001.

Teaching

Classes I taught at NC State.

Term	Class
Fa 2023	Software Testing (G), 11 students
Sp 2023	Software Engineering (G), 60 students

Classes I taught at the Federal University of Pernambuco (UFPE) in reverse-chronological order. The academic year at UFPE consists of two semesters, referred to as year.1 and year.2. The letters U and G indicate Undergraduate and Graduate-level courses, respectively. For undergraduate courses, there are two classes per week, and the duration of a class is 2h. Graduate-level courses are more flexible.

Term	Class 1	Class 2
2022.1	Software Engineering (U), 53 students	Advanced Software Testing (G), 22 students
2021.2	Software Engineering (U), 48 students	Advanced Software Testing (G), 16 students
2021.1	Software Engineering (U), 58 students	Advanced Software Testing (G), 19 students
2020.2	Compilers (U), 22 students	Software Testing (U), 27 students
2020.1	Compilers (U), 62 students	Software Testing (U), 17 students
2019.2	Compilers (U), 24 students	Software Testing (U), 34 students
2019.1	Compilers (U), 43 students	Advanced Software Testing (G), 21 students
2018.2	Compilers (U), 14 students	Advanced Software Testing (G), 15 students
2018.1	Compilers (U), 27 students	Advanced Software Testing (G), 18 students
2017.2	Compilers (U), 21 students	Advanced Software Testing (G), 34 students
2017.1	Compilers (U), 22 students	Advanced Software Testing (G), 20 students
2016.2	Compilers (U), 29 students	Advanced Software Testing (G), 28 students
2016.1 2015.2	sabbatical	
2015.1	Compilers (U), 45 students	Seminar in Software Testing (G), 18 students
2014.2	Compilers (U), 34 students	Seminar in Software Testing (G), 7 students
2014.1	Compilers (U), 35 students	Seminar in Software Testing (G), 9 students
2013.2	Compilers (U), 40 students	Seminar in Software Testing (G), 9 students
2013.1	Compilers (U), 31 students	Seminar in Software Testing (G), 12 students
2012.2	Compilers (U), 32 students	Introduction to Computing (U), 68 students
2012.1	Compilers (U), 38 students	Introduction to Computing (U), 71 students
2011.2	Compilers (U), 40 students	Introduction to Computing (U), 73 students
2011.1	Compilers (U), 38 students	Introduction to Computing (U), 70 students
2010.2	Introduction to Computing (U), 72 students	Introduction to Computing (U), 80 students
2010.1	Functional Programming (U), 15 students	Introduction to Static Analysis (G), 10 students
2009.2	Operating Systems (U), 24 students	Introduction to Computing (U), 91 students
2009.1	Introduction to Computing (U), 77 students	Introduction to Computing (U), 67 students