Mark Meretzky
334 East Fifth Street, apt. 9
New York, NY 10003-8809
Cell: (212) 473-3684
Email: mark.meretzky@gmail.com
Web: http://oit2.scps.nyu.edu/~meretzkm/
LinkedIn: https://www.linkedin.com/in/mark-meretzky-0b2b619b
Facebook: https://www.facebook.com/MarkMeretzky

Professional Skills

Fluent in C and C++ under Unix, with experience in Java on Android, Swift on Apple iOS, and miscellaneous languages such as Python, Perl, Ruby, and Lua. A veteran of 20 years of C++ programming plus six years in the compiler industry, with practice in modularizing large programs, managing complexity, and making source code easier to understand.

Also a dynamic speaker with a special interest in organizing information in tutorial form, ranging from one-hour lectures to full-semester courses. Skilled at putting himself in the learner's place, finding the structure in an amorphous body of knowledge, and creating suites of economical examples that build on each other.

Have taught courses in C, C++, Unix, Python, Android, and iOS at schools including NYU, Pace, and Columbia; at companies including Bloomberg, CitiBank, Lehman Brothers, Morgan Stanley, Scudder, and SIAC; and at conferences including Agile, DevCon5, QCon, and SANS.

Experience

Oct. 1990– **New York University School of Professional Studies** **www.sps.nyu.edu**
Adjunct Associate Professor of Information Technologies. Created SPS's programs in Unix, C++, Ruby on Rails, iOS, and Android, and also taught C, Python, Perl, Java, and JavaScript. Each evening adult ed course consisted of 10 or 12 weekly three-hour lectures attended by 30 to 50 students of wildly differing abilities. Decided which topics should not be attempted to avoid overwhelming the students; produced book-length handouts (on my website) used by other instructors; and invented and corrected rigorous homework. Award for excellence in teaching, May 1993.

Pioneered Unix education with courses on scripting, filters, regular expressions, and editing and transforming data. Courses in Unix system calls included clients and servers with TCP/IP sockets, forking, and multi-threading. C courses emphasized control structure, expression evaluation, and pointers and indirection. C++ courses built bigger data types out of smaller ones with aggregation, inheritance, and templates; presented operator overloading as the natural notation for i/o, dynamic memory allocation, and iterators; and explored the extensible architecture of the STL. The C++ courses illustrated all of these features with an extensive body of code culminating in one large, evolving video game.

iOS and Android courses concentrated on the user interfaces, underpinned by ''delegates'' in iOS and ''listeners'' in Android, and were flexible enough to appeal to beginners and advanced learners. Helped students wrestle with the IDEs (Xcode, Eclipse, and Android Studio) and language issues (converting iOS courses from Objective-C to Swift in 2014).

Nov. 2020– **InterSource Switzerland** **www.intersource.ch**
Dec. 2020 Taught advanced C++17 to programmers and physicists at CERN via Zoom. Covered inheritance and polymorphism, templates and the STL, concurrency and multithreading.

Jan. 2020– **mthree Consulting** **https://www.mthree.com**
Feb. 2020 Taught four-week intensive full stack course to prepare recent college graduates in Computer Science, Physics, and Math for careers in the financial industry. Covered front end topics such as React; back end topics such as Java; and Linux, shellscripting, and Python scripting.

Oct. 2018– **Borough of Manhattan Community College** **www.bmcc.cuny.edu**
Created and taught 270-hour Apple iOS courses in the language Swift, for students with no prior programming experience. Also taught Python and pandas.

Jul. 2015– **Crossfire Media** **www.xfiremedia.com**
Taught full-day tutorials for Android in Java and Apple iOS in Swift at the DevCon5 Web & Mobile App Developer Conferences in New York.

Apr. 2015– **Udacity** **www.udacity.com**
Jul. 2015 Worked with Google personnel to create their online Android course for beginners in XML and Java. Created content including technical definitions and cheat sheets. Supervised a graphics firm in England and Barcelona (psycle.com) through many iterations of the accompanying illustrations.

Oct. 2014– **Pace University, New York, NY** **www.pace.edu**
Feb. 2015 Created and taught a 60-hour Apple iOS course (CRN 90200) in Swift, live in the classroom and delivered to remote students via WebEx.

Jun. 2014 **Agile Conference, Las Vegas, NV** **adc-bsc-west.techwell.com**
Taught the full-day Android tutorial in XML and Java at the Agile Development Conference West in Caesars Palace.

Jun. 2013– **QCon Conference** **www.qconferences.com**
Jun. 2015 Taught full-day Android and iOS courses at five QCon conferences in New York and San Francisco. On each platform, covered the IDE (Android Studio and Xcode), the basic architecture of an app (lifecycle methods, listeners and delegates, touch sensitivity), and the user interface. iOS was in the language Objective-C in 2013, Swift thereafter.

Oct. 2005– **Andrus Planetarium, Hudson River Museum, Yonkers, NY** **www.hrm.org**
May 2009 Composed 48-minute planetarium shows with attention to pacing, momentum, and the limits of the format. Operated a manual, pre-computer (1980s era) Zeiss M101−5 projector in total darkness while lecturing, fielding questions, cuing special effects, and wielding a laser pointer. Audiences of up to 120 ranged from infants to retirees. Programmed animations in *Mathematica*, scripted astronomical simulations in Celestia using a Lua superset and wrote the first tutorial for it, and documented how the planetarium worked. Profiled in *The New York Times*, **http://www.nytimes.com/2009/02/23/nyregion/23bigcity.html**

Jan. 2000– **Bloomberg, New York, NY** **www.bloomberg.com**
Jun. 2001 Created and taught a series of three-week C and Unix courses tailored to the needs of Bloomberg personnel. The C courses concentrated on algorithms and the idiosyncratic features of the language: the derived data types (pointers, structures, and exotic combinations thereof) and dynamic memory allocation. The Unix courses concentrated on filters, shellscripting, and searching and editing with regular expressions.

Jan. 1999– **Manhattan Center for Science and Math** **www.mcsm.net**
May 1999 Taught C to high school sophomores and juniors with no previous programming experience, in an experimental afterschool program in East Harlem funded by George Soros. Created examples relevant to the chemistry and trig classes they were taking during the day.

May 1996– **System Administration and Networking Conference** **www.sans.org**
May 1999 Taught full-day Tcl, Tk, and Expect seminars to Unix system administrators at SANS96 (Washington, DC), SANS97 (Baltimore, MD), SANS98 (Monterey, CA), and SANS99 (Baltimore, MD).

Sept. 1994– **New York University College of Arts and Science** **www.nyu.edu**
Dec. 1995 Taught first- and third-semester Calculus to physics, engineering, and pre-med undergraduates in classes of up to 100 students, with teaching assistants. Calc I (V63.0021) covered differentiation and integration; Calc III (V63.0023) covered vector calculus: Stokes' Theorem, Green's Theorem, etc. Generated 2- and 3-D graphics with *Mathematica.*

Oct. 1990– **Miscellaneous consulting jobs**
Dec. 2000 Taught C, C++, and Unix courses at Citibank, Lehman Brothers, Morgan Stanley, Scudder, SIAC, and the Columbia University Division of Continuing Education.

Oct. 1988– **Tradenet, Inc., New York, NY**
May 1991 Wrote proprietary object-oriented software for Apple Macintosh II in Symantec Think C and its class library.

Oct. 1987–
July 1988

Clint Goss Consultants, Ltd., Westport, CT **www.goss.com/cgc.htm**

Consultant at SIAC (Securities Industry Automation Corporation), Inc. on a project to translate a large corpus of software from TAL (Tandem Computers' Transaction Application Language) to Tandem C. Set policy on which TAL idioms and data structures could be translated by machine and devised readable C equivalents for them; identified other constructs which had to be recoded by hand in C to be maintainable. Wrote TAL-to-C translator in *yacc* and C, running under Tandem's GUARDIAN operating system, which also flagged code that required additional hand tuning. Trained SIAC personnel to use the translator and perform remaining coding.

Nov. 1981–
May 1987

Philon, Inc., New York, NY

Project leader for developing the retargetable and portable code generator for the Philon multi-language/multi-machine compiler system. Designed and implemented the code generation data structures and algorithm in C under Unix; established and policed the software engineering methodology for keeping the language, host, and target dependencies under control; created a development environment which hid the complexity of the rest of the compiler and code generator from the programmers who wrote the machine-specific code; and trained them to use the system to build on each other's work.

Wrote concise documentation: end-user BASIC manuals and tutorials, in-house system documentation, and customer proposals analyzing the time and resources needed to retarget the Philon system for new architectures.

Developed compilers for C, CBASIC, Microsoft BASIC, Fortran, and Pascal, generating code for DEC PDP-11 and VAX-11, Intel 8086, Motorola 68000, National Semiconductor 32000, AT&T WE32100 (proprietary), and Computer Consoles 6/32 (proprietary).

Aug. 1981–
Nov. 1981

Datamat, Inc., New York, NY

Technical sales support for a computer system for the numerical control of machine tools in heavy manufacturing. Taught 40-hour courses in GTL3, Olivetti's high-level language for describing the size and shape of parts to be machined out of metal, and the sequences of moves that the cutting tools should make. Maintained the GTL3 postprocessors (code generators) for several target machine tools. On-site and remote trouble shooting, customer support.

June 1980–
Feb. 1981

Radio Shack, New York, NY **www.radioshack.com**

Created and taught 10- and 25-hour classes, beginner and advanced, in Microsoft BASIC and Zilog Z80 assembly language on the legendary TRS-80 computer. Since the assembler course was one of the few available at the time, it was general enough for people whose primary goal was to learn a different assembly language.

Education

Sept. 1988–

Graduate School of Arts and Science, New York University **www.nyu.edu/gsas/**

Part-time candidate for MS in Mathematics. Have taken Theory of Computation, Abstract Algebra, Topology, Linear Algebra, Analysis, Complex Variables.

July 1979–
June 1981

Washington Square and University College, New York University **www.nyu.edu/cas/**

BA in Computer Science, June 1981. Programmed in PL/I, Fortran, SNOBOL4, 8080 assembler. BASIC advisor, PL/I grader, Systems Programming teaching assistant. Founder's Day Award, April 1981.

Sept. 1972–
June 1974

Columbia College, New York, NY **www.college.columbia.edu**

Ancient Semitic Languages: Akkadian, Ugaritic. Dean's List, 1974.

Interests

Teaching programming, mathematics, and astronomy. First person to bicycle across the Sinai Desert (Cairo to Tel Aviv), October 1986.