

# Charlie Abzug's Home Page



**Charles Abzug, Ph.D.**  
**Visiting Professor, Department of Philosophy & Religion**  
**James Madison University, MSC 8006**  
**Cleveland Hall, Room 312**  
**61 East Grace Street**  
**Harrisonburg, VA 22807**

Cell Phone: 443-956-9424

FAX: 540-568-8072

Home Page: <http://users.cs.jmu.edu/abzugcx/public/index.htm>

E-Mail: [CharlesAbzug@Acm.org](mailto:CharlesAbzug@Acm.org), [AbzugCX@JMU.edu](mailto:AbzugCX@JMU.edu)

## ***Education:***

***S.B.:*** Massachusetts Institute of Technology (Life Sciences)

***M.S.:*** New York Medical College (Physiology with minor in Anatomy)

***Ph.D.:*** New York Medical College (Physiology with minor in Anatomy)

***M.S.:*** Johns Hopkins University (Computer Science)

***Additional Studies (in chronological order):*** Harvard University, Columbia University, George Washington University, University of Maryland Baltimore County, University of Maryland College Park

***Postdoctoral Studies:*** Rockefeller University

## ***Conference Hours***

***Reaching Me by Telephone:*** I like to be maximally accessible to all of my students. I consider my students to be anyone enrolled in any of my classes, as well as any student for whom I serve as advisor. Therefore, if you are one of my students, then you are welcome to call me on my cell phone at any time of the day or night, any day of the week. Should I happen to be either in class or in a meeting when you call, or should I be unable to answer for any other reason, you will be passed over to my voice mail. Please leave me a substantive message. That is, don't leave me just your name and number, but also inform me why you are calling. This will enable me not only to get back to you as quickly as possible, but also to be better prepared, when I do get back to you, to deal effectively with whatever you want to talk about. If you are not one of my students, then you are welcome to call me Sunday through Thursday between the hours of 0800 and 2200 Eastern Time, and on Friday between 0800 and sunset.

***Note to My Student Advisees:*** In order for me to provide you with good advice, you and I need to take

the trouble to get acquainted. Please make an appointment to come in and meet with me, so that I can get to know you as a person and also become familiar with your academic background, as well as with your personal preferences and career goals. An initial meeting with one of my student advisees usually takes between fifteen and thirty minutes. I usually take notes not only at the first, but also at all subsequent meetings. Using my notes, I can refresh my memory several weeks or months later when you send me an E-mail or call me up on the telephone or come in to see me, either to seek my advice on a problem you are having, or to ask a question. Sometimes, I get a frantic E-mail or find an agitated voice-mail message on my telephone from a student whom I have never met, who suddenly realizes that he has a serious problem, concerning which he needs advice. It is nearly impossible for me to advise a student based upon my receipt of a hasty and invariably fragmentary description of a problem coming from a complete stranger. So do invest a few minutes up front so that we can get to know each other. Then, **when** (not if) you encounter a problem, I will have a basis to provide you with sound advice. My weekly schedule listing conference hours for meeting with students, together with instructions on how to go about arranging to meet with me, are provided in the following document: [Conference Hours](#).

***My Personal Policy on Gifts from Students:*** As part of my duties as a member of university faculty, I assign course grades to students. Also, I am sometimes called on to write letters of recommendation for current or former students, to meet with agents of the FBI and of various other U.S. Government intelligence services to discuss the suitability of a present or former student for a security clearance, and to perform various other functions that require me objectively to evaluate a student's performance in schoolwork or his/her personal or professional suitability for some job, for admission to graduate or professional school, for a security clearance, or for an award. In order that my judgment on such matters not be influenced by extraneous considerations, I have adopted a strict policy of not accepting **any** gifts from present or former students. I hope you will not be insulted, but because of my strict policy in this regard, should you offer me a gift I shall decline to accept it, and should you leave one for me in my mail box or under my office door or elsewhere, then I shall **insist** on returning it to you.

## [How Did I Determine Your Grade](#)

### ***Current and Recent Courses:***

[Math/CS-227: Discrete Structures I](#). Notation and interpretation of formal logic, use of formal logic in determining the validity of arguments and in designing, evaluating, and specifying digital logic circuits. Quantified statements and their logic. Elementary number theory, direct proof, proof by counterexample, and indirect argument. Sequences, mathematical induction, and set theory. *Prerequisites: none.*

[Math/CS-228: Discrete Structures II](#). Techniques of counting, including a variety of example problems that demonstrate the utility of counting in several contexts of computer science. It progresses through combinatorial analysis, and goes on to cover discrete probability, elementary number theory and modular arithmetic, integer and fixed-point representation of numbers, as well as floating-point representation, integer or fixed-point arithmetic, logic gates and digital logic, Boolean algebra, and techniques of circuit minimization. *Prerequisite: Math/CS 228.*

[CS-350: Computer Organization](#). Students learn how a digital computer works through study of the principles of basic electronic circuits, simple logic circuits, and integrated circuits of the two principal types, combinational and sequential. The course progresses from the basic circuit level to the amalgamation of various simple circuits to produce more complex circuits of steadily increasing complexity as well as increasing levels of integration, including Small-Scale Integration or SSI, Medium-Scale Integration or MSI, Large-Scale Integration or LSI, and Very-Large-Scale Integration or VLSI. Over the course of the semester students will learn how it is possible to

construct a complete working computer through the design of such components as a multiplexor, decoder, encoder, demultiplexor, register, bus, arithmetic-logic unit, microprocessor, and memory module. The principles of hierarchical computer organization, of digital integer arithmetic, machine instruction sets, addressing modes, input/output processing, and interrupt handling are also covered, as well as the application of many of these concepts to modern personal computers. *Prerequisite: CS 240.*

[CS-450-&-CS-550: Operating Systems \(formerly CS-351\).](#) Role and importance of the operating system in the efficient and productive utilization of the modern computer. Functions and evolution of operating systems. Process control, threads, symmetric multiprocessing, and microkernels. Problems associated with concurrency, and the use of mutual exclusion, semaphores, monitors, and message passing to solve them. The problems of deadlock and starvation. Methods of memory management, including partitioning, paging, segmentation, and virtual memory. Scheduling in uniprocessor, multiprocessor, and real-time environments. General management of Input/Output operations, and in particular the problem of disk scheduling. Management of files, including file organization, file directories, file sharing, and record blocking. . *Prerequisite: CS 350.*

[CS-627: Cryptology.](#) This course provides the student with an understanding of the principles and operational aspects of cryptography and with the ability to implement major encryption protocols. These protocols provide one of the major procedures and product families used to protect communication channels used in government and private enterprises. The course deals with the design and analysis of systems that provide protection for communications and that resist cryptanalysis. Social and societal implications of current cryptographic systems will be discussed. *Prerequisites: CS 621.*

[HEBR/REL-131-&-132: Elementary Biblical Hebrew I & II.](#) This course is designed to rapidly equip the student with an ability to comprehend the Massoretic (i.e., the classical or traditional) text of the Hebrew Bible, or *TaNaKh*, as it is called in Hebrew<sup>[1]</sup>. Systematic study is undertaken of the fundamentals of Hebrew grammar as used in the *TaNaKh*, together with the rapid development of a vocabulary consisting of one hundred or more of the words most commonly appearing in it. There is a heavy emphasis on reading and pronunciation, in addition to translation. *Prerequisites: none.*

## [Student-Produced-Term-Projects](#)

### ***My Children's Home Pages:***

[Mordy](#)

[Rinah](#)

### ***My Teaching Activities of former years:***

[CS 128: Information Presentation.](#) Survey of ways to use computer software tools to locate and retrieve information, to present that information in written formats that include tables, graphs and pictures, and to integrate that information into verbal presentations that may include audio and video materials as well. *Prerequisites: none.*

[CS-139: Algorithm Development.](#) This is the introductory course that provides the student with the basic knowledge, skill, and understanding that forms the foundation for all his/her subsequent study of Computer

Science. The principal objective of this course is to instill in the novice programmer sound programming practices, starting with careful analysis of the problem to be solved by the program, and progressing through thorough design, construction, testing, and deployment of a high-quality computer program (software). *Prerequisites: none.*

## [CS-452 & CS-552: Design and Analysis of Algorithms, & Applied Complexity](#)

Theory. Algorithmics consisting of the study of the design and analysis of algorithms, is based upon of the application of mathematical techniques to the analysis of the efficiency of algorithms, and in particular to the effect upon the time to execute different algorithms of both the growth in the quantity of the input data to be acted upon and of the quality of the input data. The use of formal mathematical techniques goes back to the earliest times in the development of, and is probably the most important and certainly the most complex area of study within, the overall field of Computer Science. Algorithmics and related subject matter make extensive use of the mathematical discipline of complexity theory. *Prerequisite: Math/CS 228 and CS-240.*

## [CS-480/585A: Computer Architecture.](#)

This course builds upon the basics of Computer Organization that were previously covered in the course of that name, and delves in considerably greater depth into issues of Computer Architecture. Computer Architecture is first defined and distinguished from Computer Organization. The principles of digital logic are reviewed in depth, in conjunction with the principal axioms and theorems of Boolean Algebra. The representation of data in digital computers is covered in several aspects: Integer or Fixed-Point Representation is covered in several variants. Floating-Point Representation is discussed with emphasis on the issue of precision. Arithmetic is discussed in several of the representation schemes including addition, subtraction, multiplication, and division. Instruction-Set Architecture, Assembly, compilation, subroutine linkage, loading, and use of a stack are covered. Microprogramming is contrasted with Hardwired control. Memory hierarchy is discussed in depth, and some modern streams in Computer Architecture. *Prerequisite: CS 350 or CS 511.*

## [CS-480/585I: Information Systems Security.](#)

This course provides a solid theoretical foundation in Information Systems Security (INFOSEC), including both Computer Security and Communications Security. In addition, various terms used in the field will be defined, and the basic principles underlying INFOSEC will be laid out. Subjects to be covered include Identification and Authentication, including not just Username and Password but also other techniques, such as biometrics, and also a survey of techniques used by attackers to reveal users' passwords. Principles of access control will be discussed, including subject-oriented, object-oriented, and hybrid techniques, and also several security models, including the Bell-LaPadula (BLP) model, the Biba and Clark-Wilson models, and the Chinese Wall model. Techniques for assuring that the system enforces the stated security policy will be covered, as well as the practical implementation of security in several representative systems. Problems often encountered in system security will be discussed, as well as the evaluation of a system to determine the level of soundness with which security has been implemented. *Prerequisites: CS-350 or CS-511, and CS-450 or CS-550.*

## [CS-511: Accelerated Fundamentals of Computer Systems: Computer](#)

Organization. Students learn how a digital computer works through thorough study of the principles of operation of a simple but workable machine. They cover the principal subsystems of a computer, including the central processing unit (CPU), memory, input/output ports, communications bus, and several types of input/output devices (including keyboard and video display terminal). Number systems and various schemes for the digital representation of numbers are also discussed, as well as the principles of digital integer arithmetic, to provide insight into several different ways that arithmetic calculations can be made in digital computers. The major computer subsystems are surveyed, including the CPU, system bus and subsidiary buses, primary storage (RAM and ROM), and secondary storage (HDD and floppy drives). Basic electronics is reviewed, in order to convey an understanding of the operation of logic circuits (AND gates, OR gates, NOT gates, buffer gates, XOR gates and XNOR gates) and digital integrated circuits. Computer operations are introduced, along with the representation of computer operations in an Assembly Language. Implementation of program statements in a Higher Level Language, such as C or C++, is studied through the decomposition of each higher-level language statement into a sequence of machine or Assembly Language

instructions, and some of these are further decomposed into sequences of micro-operations. The operation of major subsystems of the computer is examined through decomposition of a subsystem into an organized set of simpler circuit elements. Additional critical subjects covered include the principles of hierarchical computer organization, machine instruction sets, addressing modes, CISC vs. RISC, input/output processing, and interrupt handling, as well as the application of many of these concepts to modern personal computers, and the boot process. The student partitions the hard drive on his/her own computer and installs the LINUX operating system into a separate bootable partition on the machine, in addition to the Windows-95/98 or Windows-NT operating system already there, in order to gain deeper insight into the boot-up process and to prepare the groundwork for the study of Operating Systems. *Prerequisite: CS 510.*

**[CS-585F: Computer-Related Law and Computer Forensics.](#)** Basic principles of the American legal environment: the United States constitution, various court systems and jurisdictional issues, and handling and resolution within the legal system of conflicts (lawsuits). Aspects of the legal environment particularly relevant to computers. Civil responsibilities and liabilities of hardware vendors, software vendors, and computer service providers. Criminal liability. Intellectual property issues. Identification, preservation, extraction, and interpretation of information present on computer storage media to analyze what transpired in a computer security incident, to determine the identities of the perpetrators, and to provide evidence admissible in court. Techniques used to determine whether any criminal activity might have occurred and to acquire the evidence necessary for prosecution while avoiding alteration of or damage to the original data. Computer viruses, Trojan Horse software, and other malware (hostile code). Various software tools useful for forensic examination. *Prerequisite/Corequisite: CS 350 or CS-511.*

**[CS-621: Trusted Systems.](#)** Definition of a "Trusted System", and considerations pertaining to the design, evaluation, certification and accreditation of trusted systems, including hardware considerations, software considerations such as developmental controls, validation/verification, assured distribution and other assurance issues. Implementation, configuration management, and systems administration of trusted systems. Trusted applications and trusted database issues. Importance of aggressive monitoring and of setting traps for the intruder. Importance of the understanding of both the psychology and the successful *modus vivendi* of the attacker to generating and maintaining a powerful defense. *Prerequisites: CS 510, 511, 512, 550, 555, 574, and CS-620.*

**[CS-622: Policy, Legal Issues, and Ethics.](#)** Description of the current regulatory and legal environment in computer security. Contemporary issues in computer security regulations and laws, including property law, criminal law, international law, and telecommunications. Ethical issues in an information architecture and their relationship to legal standards. *Prerequisites: CS 620.*

This is Iteration 7.0 of my home page. Please address comments/suggestions to: [CharlesAbzug@ACM.org](mailto:CharlesAbzug@ACM.org).

---

<sup>[1]</sup> Pronunciation: *Tah—Nach*?. *Tah* rhymes with *Bah*, and *Nach* rhymes with *Bach*, as in *Johann Sebastian Bach*.