webcast.berkeley

courses events

Computer Science | Any Semesters

Popular Courses

Integrative Biology 131 - Lecture 01: Organization of Body
Professor Marian Diamond

Physics 10 - Lecture 01: Atoms and Heat
Professor Richard A. Muller

SIMS 141 - Search, Google, and Life
Sergey Brin - Google

Psychology 1 - General Psychology - Lecture 1
Professor John F. Kihlstrom

PACS 164A: Introduction to Nonviolence
Professor Michael Nagler

Computer Science 10, 001 - Spring 2015
DANIEL GARCIA

The Beauty and Joy of Computing - An introduction to the beauty and joy of computing. The history, social implications, great principles, and future of computing. Beautiful applications that have changed the world. How computing empowers discovery and progress in other fields. Relevance of computing to the student and society will be emphasized.

Computer Science 162, 001 - Spring 2015
JOHN KUBIATOWICZ

Operating Systems and System Programming - Basic concepts of operating systems and system programming. Utility programs, subsystems, multiple-program systems. Processes, interprocess communication, and synchronization. Memory allocation, segmentation, paging. Loading and linking, libraries. Resource allocation, scheduling, performance evaluation. File systems, storage devices, I/O systems. Protection, security, and privacy.

Computer Science 169, 001 - Spring 2015
ARMANDO FOX

Software Engineering - Ideas and techniques for designing, developing, and modifying large software systems. Function-oriented and object-oriented modular design techniques, designing for re-use and maintainability. Specification and documentation.

Verification and validation. Cost and quality metrics and estimation. Project team organization and management. Students will work in teams on a substantial programming project.

Computer Science 170, 001 - Spring 2015

PRASAD RAGHAVENDRA, CHRISTOS H PAPADIMITRIOU

Efficient Algorithms and Intractable Problems - Concept and basic techniques in the design and analysis of algorithms; models of computation; lower bounds; algorithms for optimum search trees, balanced trees and UNION-FIND algorithms; numerical and algebraic algorithms; combinatorial algorithms. Turing machines, how to count steps, deterministic and nondeterministic Turing machines, NP-completeness. Unsolvable and intractable problems.

Computer Science 186, 001 - Spring 2015

JOSEPH HELLERSTEIN

Introduction to Database Systems - Access methods and file systems to facilitate data access. Hierarchical, network, relational, and object-oriented data models. Query languages for models. Embedding query languages in programming languages. Database services including protection, integrity control, and alternative views of data. High-level interfaces including application generators, browsers, and report writers. Introduction to transaction processing. Database system implementation to be done as term project.

Computer Science 188, 001 - Spring 2015

PIETER ABBEEL, DAN KLEIN

Basic ideas and techniques underlying the design of intelligent computer systems. Topics include heuristic search, problem solving, game playing, knowledge representation, logical inference, planning, reasoning under uncertainty, expert systems, learning, perception, language understanding.

Computer Science 195, 001 - Spring 2015

JOHN S. DENERO

Social Implications of Computer Technology - Topics include electronic community; the changing nature of work; technological risks; the information economy; intellectual property; privacy; artificial intelligence and the sense of self; pornography and censorship; professional ethics. Students will lead discussions on additional topics.

Computer Science 198, 032 - Spring 2015

BJOERN HARTMANN

Web Design Decal.

Computer Science 61A, 001 - Spring 2015

JOHN S. DENERO

The Structure and Interpretation of Computer Programs - Introduction to programming and computer science. This course exposes students to techniques of abstraction at several levels: (a) within a programming language, using higher-order functions, manifest types, data-directed programming, and message-passing; (b) between programming languages, using functional and rule-based languages as examples. It also relates these techniques to the practical problems of implementation of languages and algorithms on a von Neumann machine. There are several significant programming projects, programmed in a dialect of the LISP language.

Computer Science 61B, 002 - Spring 2015

JOSHUA A. HUG

Data Structrues - Fundamental dynamic data structures, including linear lists, queues, trees, and other linked structures; arrays strings, and hash tables. Storage management. Elementary principles of software engineering. Abstract data types. Algorithms for sorting and searching. Introduction to the Java programming

language.

Computer Science 61C, 001 - Spring 2015

VLADIMIR STOJANOVIC. KRSTE ASANOVIC

Machine Structures - The internal organization and operation of digital computers. Machine architecture, support for high-level languages (logic, arithmetic, instruction sequencing) and operating systems (I/O, interrupts, memory management, process switching). Elements of computer logic design. Tradeoffs involved in fundamental architectural design decisions.

Computer Science 70, 001 - Spring 2015

UMESH VAZIRANI

Discrete Mathematics and Probability Theory - Logic, infinity, and induction; applications include undecidability and stable marriage problem. Modular arithmetic and GCDs; applications include primality testing and cryptography. Polynomials; examples include error correcting codes and interpolation. Probability including sample spaces, independence, random variables, law of large numbers; examples include load balancing, existence arguments, Bayesian inference.

Computer Science 98, 052 - Spring 2015

JOHN S. DENERO

Additional Topics on the Structure and Interpretation of Computer Programs. - The course must be taken concurrently with Computer Science 61A and will cover additional topics and examples related to Computer Science 61A.

Computer Science 10, 001 - Fall 2014

GERALD FRIEDLAND

The Beauty and Joy of Computing - An introduction to the beauty and joy of computing. The history, social implications, great principles, and future of computing. Beautiful

applications that have changed the world. How computing empowers discovery and progress in other fields. Relevance of computing to the student and society will be emphasized.

Computer Science 162, 001 - Fall 2014

DAVID CULLER

Operating Systems and System Programming - Basic concepts of operating systems and system programming. Utility programs, subsystems, multiple-program systems. Processes, interprocess communication, and synchronization. Memory allocation, segmentation, paging. Loading and linking, libraries. Resource allocation, scheduling, performance evaluation. File systems, storage devices, I/O systems. Protection, security, and privacy.

Computer Science 170, 001 - Fall 2014

DAVID WAGNER

Efficient Algorithms and Intractable Problems - Concept and basic techniques in the design and analysis of algorithms; models of computation; lower bounds; algorithms for optimum search trees, balanced trees and UNION-FIND algorithms; numerical and algebraic algorithms; combinatorial algorithms. Turing machines, how to count steps, deterministic and nondeterministic Turing machines, NP-completeness. Unsolvable and intractable problems.

Computer Science 61A, 001 - Fall 2014

JOHN S. DENERO

The Structure and Interpretation of Computer Programs - Introduction to programming and computer science. This course exposes students to techniques of abstraction at several levels: (a) within a programming language, using higher-order functions, manifest types, data-directed programming, and message-passing; (b) between programming languages, using functional and rule-based languages as examples. It also relates these techniques to the practical problems of implementation

of languages and algorithms on a von Neumann machine. There are several significant programming projects, programmed in a dialect of the LISP language.

Computer Science 61B, 001 - Fall 2014

PAUL HILFINGER

Data Structrues - Fundamental dynamic data structures, including linear lists, queues, trees, and other linked structures; arrays strings, and hash tables. Storage management. Elementary principles of software engineering. Abstract data types. Algorithms for sorting and searching. Introduction to the Java programming language.

Computer Science 61B, 003 - Fall 2014

JOSHUA A. HUG

Data Structrues - Fundamental dynamic data structures, including linear lists, queues, trees, and other linked structures; arrays strings, and hash tables. Storage management. Elementary principles of software engineering. Abstract data types. Algorithms for sorting and searching. Introduction to the Java programming language.

Computer Science 61C, 001 - Fall 2014

SHIMON MICHAEL LUSTIG, DANIEL GARCIA

Machine Structures - The internal organization and operation of digital computers. Machine architecture, support for high-level languages (logic, arithmetic, instruction sequencing) and operating systems (I/O, interrupts, memory management, process switching). Elements of computer logic design. Tradeoffs involved in fundamental architectural design decisions.

Computer Science 70, 001 - Fall 2014

ANANT SAHAI

Discrete Mathematics and Probability Theory - Logic, infinity, and induction;

applications include undecidability and stable marriage problem. Modular arithmetic and GCDs; applications include primality testing and cryptography. Polynomials; examples include error correcting codes and interpolation. Probability including sample spaces, independence, random variables, law of large numbers; examples include load balancing, existence arguments, Bayesian inference.

Computer Science C149, 001 - Fall 2014

EDWARD A. LEE, ALBERTO SANGIOVANNI-VINCENTELLI

Introduction to Embedded Systems - This course introduces students to the basics of models, analysis tools, and control for embedded systems operating in real time. Students learn how to combine physical processes with computation. Topics include models of computation, control, analysis and verification, interfacing with the physical world, mapping to platforms, and distributed embedded systems. The course has a strong laboratory component, with emphasis on a semester-long sequence of projects.

Computer Science 10, 001 - Spring 2014

DANIEL GARCIA

The Beauty and Joy of Computing - An introduction to the beauty and joy of computing. The history, social implications, great principles, and future of computing. Beautiful applications that have changed the world. How computing empowers discovery and progress in other fields. Relevance of computing to the student and society will be emphasized.

Computer Science 162, 001 - Spring 2014

ANTHONY D. JOSEPH

Operating Systems and System Programming - Basic concepts of operating systems and system programming. Utility programs, subsystems, multiple-program systems. Processes, interprocess communication, and synchronization.

Memory allocation, segmentation, paging. Loading and linking, libraries. Resource allocation, scheduling, performance evaluation. File systems, storage devices, I/O systems. Protection, security, and privacy.

Computer Science 61A, 001 - Spring 2014

PAUL HILFINGER

The Structure and Interpretation of Computer Programs - Introduction to programming and computer science. This course exposes students to techniques of abstraction at several levels: (a) within a programming language, using higher-order functions, manifest types, data-directed programming, and message-passing; (b) between programming languages, using functional and rule-based languages as examples. It also relates these techniques to the practical problems of implementation of languages and algorithms on a von Neumann machine. There are several significant programming projects, programmed in a dialect of the LISP language.

Computer Science 61B, 001 - Spring 2014

JONATHAN SHEWCHUK

Data Structrues - Fundamental dynamic data structures, including linear lists, queues, trees, and other linked structures; arrays strings, and hash tables. Storage management. Elementary principles of software engineering. Abstract data types. Algorithms for sorting and searching. Introduction to the Java programming language.

Computer Science 61C, 001 - Spring 2014

DANIEL GARCIA

Machine Structures - The internal organization and operation of digital computers. Machine architecture, support for high-level languages (logic, arithmetic, instruction sequencing) and operating systems (I/O, interrupts, memory management, process switching). Elements of computer logic design. Tradeoffs involved in fundamental

architectural design decisions.

Computer Science 162, 001 - Fall 2013

ANTHONY D. JOSEPH

Operating Systems and System Programming - Basic concepts of operating systems and system programming. Utility programs, subsystems, multiple-program systems. Processes, interprocess communication, and synchronization. Memory allocation, segmentation, paging. Loading and linking, libraries. Resource allocation, scheduling, performance evaluation. File systems, storage devices, I/O systems. Protection, security, and privacy.

Computer Science 169, 001 - Fall 2013

DAVID PATTERSON, ARMANDO FOX

Software Engineering - Ideas and techniques for designing, developing, and modifying large software systems. Function-oriented and object-oriented modular design techniques, designing for re-use and maintainability. Specification and documentation. Verification and validation. Cost and quality metrics and estimation. Project team organization and management. Students will work in teams on a substantial programming project.

Computer Science 170, 001 - Fall 2013

SATISH B RAO

Efficient Algorithms and Intractable Problems - Concept and basic techniques in the design and analysis of algorithms; models of computation; lower bounds; algorithms for optimum search trees, balanced trees and UNION-FIND algorithms; numerical and algebraic algorithms; combinatorial algorithms. Turing machines, how to count steps, deterministic and nondeterministic Turing machines, NP-completeness. Unsolvable and intractable problems.

Computer Science 294, 073 - Fall 2013

PHILLIP COLELLA

Special Topics - Software Engineering for Scientific Computing

---

Computer Science 61A, 001 - Fall 2013

JOHN S. DENERO

The Structure and Interpretation of Computer Programs - Introduction to programming and computer science. This course exposes students to techniques of abstraction at several levels: (a) within a programming language, using higher-order functions, manifest types, data-directed programming, and message-passing; (b) between programming languages, using functional and rule-based languages as examples. It also relates these techniques to the practical problems of implementation of languages and algorithms on a von Neumann machine. There are several significant programming projects, programmed in a dialect of the LISP language.

Computer Science 61B, 001 - Fall 2013

PAUL HILFINGER

Data Structrues - Fundamental dynamic data structures, including linear lists, queues, trees, and other linked structures; arrays strings, and hash tables. Storage management. Elementary principles of software engineering. Abstract data types. Algorithms for sorting and searching. Introduction to the Java programming language.

Computer Science 70, 001 - Fall 2013

UMESH VAZIRANI

Discrete Mathematics and Probability Theory - Logic, infinity, and induction; applications include undecidability and stable marriage problem. Modular arithmetic and GCDs; applications include primality testing and cryptography. Polynomials; examples include error correcting codes and interpolation. Probability including sample spaces, independence, random variables, law of large numbers; examples include load balancing,

existence arguments, Bayesian inference.

Computer Science 10, 001 - Spring 2013

DANIEL GARCIA

The Beauty and Joy of Computing - An introduction to the beauty and joy of computing. The history, social implications, great principles, and future of computing. Beautiful applications that have changed the world. How computing empowers discovery and progress in other fields. Relevance of computing to the student and society will be emphasized.

Computer Science 162, 001 - Spring 2013

ANTHONY D. JOSEPH

Operating Systems and System Programming - Basic concepts of operating systems and system programming. Utility programs, subsystems, multiple-program systems. Processes, interprocess communication, and synchronization. Memory allocation, segmentation, paging. Loading and linking, libraries. Resource allocation, scheduling, performance evaluation. File systems, storage devices, I/O systems. Protection, security, and privacy.

Computer Science 194, 024 - Spring 2013

JOHN KUBIATOWICZ

Special Topics - Advanced Operating Systems Structures and Implementation

---

Computer Science 61A, 001 - Spring 2013

AMIR ASHRAF KAMIL

The Structure and Interpretation of Computer Programs - Introduction to programming and computer science. This course exposes students to techniques of abstraction at several levels: (a) within a programming language, using higher-order functions, manifest types, data-directed programming, and message-passing; (b) between programming languages, using functional and rule-

based languages as examples. It also relates these techniques to the practical problems of implementation of languages and algorithms on a von Neumann machine. There are several significant programming projects, programmed in a dialect of the LISP language.

Computer Science 61B, 001 - Spring 2013

JONATHAN SHEWCHUK

Data Structrues - Fundamental dynamic data structures, including linear lists, queues, trees, and other linked structures; arrays strings, and hash tables. Storage management. Elementary principles of software engineering. Abstract data types. Algorithms for sorting and searching. Introduction to the Java programming language.

Computer Science 61C, 001 - Spring 2013

DANIEL GARCIA

Machine Structures - The internal organization and operation of digital computers. Machine architecture, support for high-level languages (logic, arithmetic, instruction sequencing) and operating systems (I/O, interrupts, memory management, process switching). Elements of computer logic design. Tradeoffs involved in fundamental architectural design decisions.

Computer Science 10, 001 - Fall 2012

DANIEL GARCIA

The Beauty and Joy of Computing - An introduction to the beauty and joy of computing. The history, social implications, great principles, and future of computing. Beautiful applications that have changed the world. How computing empowers discovery and progress in other fields. Relevance of computing to the student and society will be emphasized.

Computer Science 162, 001 - Fall 2012

ION STOICA

Operating Systems and System Programming - Basic concepts of operating systems and system programming. Utility programs, subsystems, multiple-program systems. Processes, interprocess communication, and synchronization. Memory allocation, segmentation, paging. Loading and linking, libraries. Resource allocation, scheduling, performance evaluation. File systems, storage devices, I/O systems. Protection, security, and privacy.

Computer Science 169, 001 - Fall 2012

DAVID PATTERSON, ARMANDO FOX

Software Engineering - Ideas and techniques for designing, developing, and modifying large software systems. Function-oriented and object-oriented modular design techniques, designing for re-use and maintainability. Specification and documentation. Verification and validation. Cost and quality metrics and estimation. Project team organization and management. Students will work in teams on a substantial programming project.

Computer Science 184, 001 - Fall 2012

RAVI RAMAMOORTHI

Foundations of Computer Graphics - Techniques of modeling objects for the purpose of computer rendering: boundary representations, constructive solids geometry, hierarchical scene descriptions. Mathematical techniques for curve and surface representation. Basic elements of a computer graphics rendering pipeline; architecture of modern graphics display devices. Geometrical transformations such as rotation, scaling, translation, and their matrix representations. Homogeneous coordinates, projective and perspective transformations. Algorithms for clipping, hidden surface removal, rasterization, and anti-aliasing. Scan-line based and ray-based rendering algorithms. Lighting models for reflection, refraction, transparency.

Computer Science 61A, 001 - Fall 2012

JOHN S. DENERO

The Structure and Interpretation of Computer Programs - Introduction to programming and computer science. This course exposes students to techniques of abstraction at several levels: (a) within a programming language, using higher-order functions, manifest types, data-directed programming, and message-passing; (b) between programming languages, using functional and rule-based languages as examples. It also relates these techniques to the practical problems of implementation of languages and algorithms on a von Neumann machine. There are several significant programming projects, programmed in a dialect of the LISP language.

Computer Science 61B, 001 - Fall 2012

PAUL HILFINGER

Data Structrues - Fundamental dynamic data structures, including linear lists, queues, trees, and other linked structures; arrays strings, and hash tables. Storage management. Elementary principles of software engineering. Abstract data types. Algorithms for sorting and searching. Introduction to the Java programming language.

Computer Science 70, 001 - Fall 2012

UMESH VAZIRANI

Discrete Mathematics and Probability Theory - Logic, infinity, and induction; applications include undecidability and stable marriage problem. Modular arithmetic and GCDs; applications include primality testing and cryptography. Polynomials; examples include error correcting codes and interpolation. Probability including sample spaces, independence, random variables, law of large numbers; examples include load balancing, existence arguments, Bayesian inference.

Computer Science 10, 001 - Spring 2012

DANIEL GARCIA

The Beauty and Joy of Computing - An introduction to the beauty and joy of computing. The history, social implications, great principles, and future of computing. Beautiful applications that have changed the world. How computing empowers discovery and progress in other fields. Relevance of computing to the student and society will be emphasized.

Computer Science 162, 001 - Spring 2012

ANTHONY D. JOSEPH, ION STOICA

Operating Systems and System Programming - Basic concepts of operating systems and system programming. Utility programs, subsystems, multiple-program systems. Processes, interprocess communication, and synchronization. Memory allocation, segmentation, paging. Loading and linking, libraries. Resource allocation, scheduling, performance evaluation. File systems, storage devices, I/O systems. Protection, security, and privacy.

Computer Science 164, 001 - Spring 2012

RASTISLAV BODIK

Programming Languages and Compilers - Survey of programming languages. The design of modern programming languages. Principles and techniques of scanning, parsing, semantic analysis, and code generation. Implementation of compilers, interpreters, and assemblers. Overview of run-time organization and error handling.

Computer Science 169, 001 - Spring 2012

DAVID PATTERSON, ARMANDO FOX, KOUSHIK SEN

Software Engineering - Ideas and techniques for designing, developing, and modifying large software systems. Function-oriented and object-oriented modular design techniques, designing for re-use and maintainability. Specification and documentation. Verification and validation. Cost and quality metrics and estimation. Project

team organization and management. Students will work in teams on a substantial programming project.

Computer Science 170, 001 - Spring 2012

CHRISTOS H PAPADIMITRIOU, SATISH B RAO

Efficient Algorithms and Intractable Problems - Concept and basic techniques in the design and analysis of algorithms; models of computation; lower bounds; algorithms for optimum search trees, balanced trees and UNION-FIND algorithms; numerical and algebraic algorithms; combinatorial algorithms. Turing machines, how to count steps, deterministic and nondeterministic Turing machines, NP-completeness. Unsolvable and intractable problems.

Computer Science 188, 001 - Spring 2012

PIETER ABBEEL

Basic ideas and techniques underlying the design of intelligent computer systems. Topics include heuristic search, problem solving, game playing, knowledge representation, logical inference, planning, reasoning under uncertainty, expert systems, learning, perception, language understanding.

Computer Science 61A, 001 - Spring 2012

PAUL HILFINGER

The Structure and Interpretation of Computer Programs - Introduction to programming and computer science. This course exposes students to techniques of abstraction at several levels: (a) within a programming language, using higher-order functions, manifest types, data-directed programming, and message-passing; (b) between programming languages, using functional and rule-based languages as examples. It also relates these techniques to the practical problems of implementation of languages and algorithms on a von Neumann machine. There are several significant programming projects, programmed in a dialect of the LISP language.

Computer Science C149, 001 - Spring 2012

EDWARD A. LEE

Introduction to Embedded Systems - This course introduces students to the basics of models, analysis tools, and control for embedded systems operating in real time. Students learn how to combine physical processes with computation. Topics include models of computation, control, analysis and verification, interfacing with the physical world, mapping to platforms, and distributed embedded systems. The course has a strong laboratory component, with emphasis on a semester-long sequence of projects.

Computer Science 10, 001 - Fall 2011

DANIEL GARCIA

An introduction to the beauty and joy of computing. The history, social implications, great principles, and future of computing. Beautiful applications that have changed the world. How computing empowers discovery and progress in other fields. Relevance of computing to the student and society will be emphasized.

Computer Science 162, 001 - Fall 2011

ANTHONY D. JOSEPH, ION STOICA

Basic concepts of operating systems and system programming. Utility programs, subsystems, multiple-program systems. Processes, interprocess communication, and synchronization. Memory allocation, segmentation, paging. Loading and linking, libraries. Resource allocation, scheduling, performance evaluation. File systems, storage devices, I/O systems. Protection, security, and privacy.

Computer Science 188, 001 - Fall 2011

DAN KLEIN

Basic ideas and techniques underlying the design of intelligent computer systems. Topics include heuristic search, problem solving, game playing,

knowledge representation, logical inference, planning, reasoning under uncertainty, expert systems, learning, perception, language understanding.

Computer Science 61B, 001 - Fall 2011

PAUL HILFINGER

Fundamental dynamic data structures, including linear lists, queues, trees, and other linked structures; arrays strings, and hash tables. Storage management. Elementary principles of software engineering. Abstract data types. Algorithms for sorting and searching. Introduction to the Java programming language.

Computer Science 61C, 001 - Fall 2011

DANIEL GARCIA, MICHAEL FRANKLIN

Machine Structures - The internal organization and operation of digital computers. Machine architecture, support for high-level languages (logic, arithmetic, instruction sequencing) and operating systems (I/O, interrupts, memory management, process switching). Elements of computer logic design. Tradeoffs involved in fundamental architectural design decisions.

Computer Science 10, 001 - Spring 2011

DANIEL GARCIA

An introduction to the beauty and joy of computing. The history, social implications, great principles, and future of computing. Beautiful applications that have changed the world. How computing empowers discovery and progress in other fields. Relevance of computing to the student and society will be emphasized.

Computer Science 164, 001 - Spring 2011

PAUL HILFINGER

Programming Languages and Compilers - Survey of programming languages. The design of modern programming languages. Principles and techniques of scanning, parsing,

semantic analysis, and code generation. Implementation of compilers, interpreters, and assemblers. Overview of run-time organization and error handling.

Computer Science 61A, 001 - Spring 2011

BRIAN HARVEY

The Structure and Interpretation of Computer Programs

Computer Science 70, 001 - Spring 2011

JAMES W. DEMMEL

Discrete Mathematics and Probability Theory

Computer Science C149, 001 - Spring 2011

SANJIT SESHIA, EDWARD A. LEE

Introduction to Embedded Systems - This course introduces students to the basics of models, analysis tools, and control for embedded systems operating in real time. Students learn how to combine physical processes with computation. Topics include models of computation, control, analysis and verification, interfacing with the physical world, mapping to platforms, and distributed embedded systems. The course has a strong laboratory component, with emphasis on a semester-long sequence of projects.

Computer Science 10, 001 - Fall 2010

BRIAN HARVEY, DANIEL GARCIA

The Beauty and Joy of Computing

Computer Science 162, 001 - Fall 2010

JOHN KUBIATOWICZ

Operating Systems and System Programming

Computer Science 169, 001 - Fall 2010

ARMANDO FOX

Software Engineering - Ideas and techniques for designing, developing, and modifying large software systems. Function-oriented and object-oriented modular design techniques, designing for re-use and maintainability. Specification and documentation. Verification and validation. Cost and quality metrics and estimation. Project team organization and management. Students will work in teams on a substantial programming project.

Computer Science 61A, 001 - Fall 2010

BRIAN HARVEY

The Structure and Interpretation of Computer Programs

Computer Science 162 - Spring 2010

ION STOICA

Operating Systems and System Programming

Computer Science 61A, 001 - Spring 2010

SATISH B RAO, BRIAN HARVEY

The Structure and Interpretation of Computer Programs

Computer Science 61C - Spring 2010

DANIEL GARCIA

Machine Structures

Computer Science 162 - Fall 2009

JOHN KUBIATOWICZ

Operating Systems and System Programming

Computer Science 61A - Fall 2009
BRIAN HARVEY

The Structure and Interpretation of Computer Programs

Computer Science 61B - Fall 2009
PAUL HILFINGER

Data Structures

Computer Science 61CL - Fall 2009
DAVID CULLER

Machine Structures

Computer Science 162 - Fall 2008
JOHN KUBIATOWICZ

Operating Systems and System Programming

Computer Science 61A - Fall 2008
BRIAN HARVEY

The Structure and Interpretation of Computer Programs

Computer Science 61B - Fall 2008
PAUL HILFINGER

Data Structures

Computer Science 61CL - Fall 2008
DAVID CULLER, MICHAEL J. CLANCY

Machine Structures
