CSE 200: Computer Assisted Problem Solving for Business

Description

Solving business related problems using and integrating productivity software tools. Emphasis on designing spreadsheet solutions to solve problems and using database queries to extract information.

Level, Credits, Class Time Distribution, Prerequisites

Level	Credits	Class Time Distribution	Prerequisites
U	5	4 1-hr lec, 1 2-hr lab	Math 116, 130, or 148

Quarters Offered

- Au, Wi, Sp, Su

General Information, Exclusions, Cross-listings, etc.

- Not open to students who have credit for CSE 101

Intended Learning Outcomes

- Be familiar with computer basics - hardware, software, OS, and communications.
- Master solving problems using simple spreadsheet formulas, functions and relative/absolute cell referencing.
- Be familiar with using Boolean logical functions and constructions - AND, OR, NOT, none-of and only constructs, IF’s and nested IF’s - in a spreadsheet.
- Be familiar with using reference functions (VLookup, HLookup) in a spreadsheet.
- Be familiar with using financial functions (FV,PV,RATE,PMT,NPER) in a spreadsheet.
- Be familiar with relational databases and database tables.
- Be familiar with querying a database using Access QBE grids to select/filter records, sort, aggregate fields, calculate fields.
- Be familiar with querying a database with multiple tables using inner and outer joins.
- Be familiar with how the internet works, protocol layers, and a setting up a simple webpage.

Texts and Other Course Materials

- *Course Technology Bundle (includes Excel New Perspectives, Shelly Cashman Access & PowerPoint and SAM CD)*
- *Course Notes* - Debra Gross (available from COP-EZ/Tuttle)

Topics

Number of Hours	Topic
2	Computer basics
5	Spreadsheet basics - formulas, functions, cell addressing
5	Boolean logic functions
2	Solving more complex problems including unit conversions and multiple worksheet solutions

3	Using financial functions
3	Using reference functions
2	Introduction to relational databases and database tables
4	Simple queries
3	Using queries with inner joins
3	Solving problems requiring multiple queries and outer joins
1	Database features of Excel
3	Internet topics - protocols and setting up a webpage
2	Using MS PowerPoint; object linking

Representative Lab Assignments

- Lab 1 - Logging on the system, using e-mail and web browsers
- Lab 2- Writing formulas in Excel - relative/absolute cell referencing and simple functions
- Lab 3 - Relational operators and Boolean logical functions
- Lab 4 - Using multiple worksheets in a workbook, solving problems with units
- Lab 5 - Solving large problems - spreadsheet design, reference functions, financial functions
- Lab 6 - PowerPoint
- Lab 7 - Using MS Access tables and simple queries
- Lab 8 - Writing queries with joins
- Lab 9 - Project integration
- Lab 10 - Creating a simple webpage

Grades

Labs	20%
Quizzes	20%
Midterm	25%
Final Exam	35%

Relationship to BS-CSE Program Outcomes/EC 2000 Criterion 3 Outcomes

a	b	c	d	e	f	g	h	i	j	k
***	**	***		**		*		**		***

Course Coordinator: Debra Gross

Department of Computer Science and Engineering
395 Dreese Laboratories
2015 Neil Avenue
Columbus, OH 43210-1277

CSE 201: Elementary Computer Programming

Description

Introduction to computer programming and to problem solving techniques using computer programs; programming lab experience.

Level, Credits, Class Time Distribution, Prerequisites

Level	Credits	Class Time Distribution	Prerequisites
U	4	3 1-hr lec, 1 2-hr lab	Mathematics placement level R or higher; or Math 075 or higher

Quarters Offered

- Au, Wi, Sp, Su

General Information, Exclusions, Cross-listings, etc.

- Not open to students with credit for CSE 221, CSE 202 or En Graph 167.
- Java is taught.

Intended Learning Outcomes

- Master using basic coding features provided by high-level imperative programming languages.
- Master writing computer programs to implement given simple algorithms.
- Be familiar with analyzing simple real-life problems and choosing appropriate algorithms for their solution.
- Be familiar with using basic data structures such as arrays in simple programs.
- Be familiar with using methods and classes to help produce well-structured programs.
- Be familiar with reading and programming for API’s.
- Be familiar with designing simple text-oriented user interfaces.
- Be familiar with working in a window-based computing environment.
- Be exposed to the services provided by an operating system.
- Be exposed to the virtual machine model of modern computer systems.
- Be exposed to data abstraction concepts and other more advanced programming ideas.

Texts and Other Course Materials

- *Java: An Introduction to Computer Science & Programming (3rd Edition), Pearson Prentice Hall, 2004, ISBN 0-13-101378-5.* - Savitch, W.,
- *CSE 201 Course Notes, OSU Reprographics, 2005.*

Topics

Number of Hours	Topic
4	Course introduction and basic concepts
4	Primitive types and expressions; String; basic I/O
8	Flow of control and Boolean expressions
5	Defining methods
4	Arrays

4	Basic exception handling and standard Java I/O
7	Classes and objects
4	Midterm and exam reviews

Representative Lab Assignments

- Environment walkthrough
- Primitive types, assignment, arithmetic expressions, simple I/O
- Control structures
- Methods
- Arrays
- Standard I/O
- Classes and objects

Grades

Midterm Exam	20%
Final Exam	30%
Homework Assignments	10%
Lab Assignments	35%
Class Participation	5%

Relationship to BS-CSE Program Outcomes/EC 2000 Criterion 3 Outcomes

a	b	c	d	e	f	g	h	i	j	k
*		**	*		*					

Course Coordinator: Paolo Bucci

Department of Computer Science and Engineering
395 Dreese Laboratories
2015 Neil Avenue
Columbus, OH 43210-1277

CSE 202: Introduction to Programming and Algorithms for Engineers and Scientists

Description

Introduction to computer programming and to problem solving techniques using computer programs with applications in engineering and the physical sciences; algorithm development; programming lab experience.

Level, Credits, Class Time Distribution, Prerequisites

Level	Credits	Class Time Distribution	Prerequisites
U	4	3 1-hr lec, 1 2-hr lab	Math 151

Quarters Offered

- Au, Wi, Sp

General Information, Exclusions, Cross-listings, etc.

Intended Learning Outcomes

- Be familiar with using basic C++ constructs: declarations and various statements including loops and conditionals.
- Be familiar with using C++ functions.
- Be familiar with using C++ arrays.
- Be exposed to using C++ structures and classes.
- Be exposed to using pointers.
- Be exposed to using file input/output.

Texts and Other Course Materials

- *C++ for Engineers and Scientists* - Bronson, Gary J.

Topics

Number of Hours	Topic
5	Structure of simple C++ programs
7	Control structures: conditionals and loops
7	C++ functions
6	Arrays
6	Structures and classes
3	Pointers
3	File I/O
3	Midterm and review

Representative Lab Assignments

- Using Unix

- Writing a simple C++ program using conditionals and (non-nested) loops
- Writing a C++ program using arrays for solving a simple scientific problem
- Program using multiple functions
- Simple program using a simple class

Grades

Programming assignments	30%
Midterm	30%
Finals	30%
Labs (closed labs)	10%

Relationship to BS-CSE Program Outcomes/EC 2000 Criterion 3 Outcomes

a	b	c	d	e	f	g	h	i	j	k
**	*	**		**		*		*		***

Course Coordinator: H David Mathias

Department of Computer Science and Engineering
395 Dreese Laboratories
2015 Neil Avenue
Columbus, OH 43210-1277

CSE 214: Data Structures for Information Systems

Description

Subroutines and modular programming; searching; basic data structures; recursion; introduction to sequential files.

Level, Credits, Class Time Distribution, Prerequisites

Level	Credits	Class Time Distribution	Prerequisites
U	4	3 cl, 1 3-hr lab	201

Quarters Offered

- Au, Wi, Sp, Su

General Information, Exclusions, Cross-listings, etc.

Intended Learning Outcomes

- Be familiar with modular design and structured programming techniques.
- Be familiar with commonly used data structures.
- Be familiar with how to design and implement abstract data types.
- Be familiar with sequential file I/O.

Texts and Other Course Materials

- *Lecture Notes* - Instructor

Topics

Number of Hours	Topic
8	Object-oriented programming
4	Recursion
8	Sorting and binary search
6	Linked lists
3	Stacks
3	Queues
4	Binary trees
4	Quizzes, exams, and review

Representative Lab Assignments

- Manipulating strings and file I/O
- Towers of Hanoi using recursion
- Sorting and recursion
- Linked lists and databases
- Trees and traversals

Grades

Homeworks	10%
Labs	25%
Quizzes	10%
Midterm Exam	20%
Final Exam	35%

Relationship to BS-CSE Program Outcomes/EC 2000 Criterion 3 Outcomes

a	b	c	d	e	f	g	h	i	j	k
*		**			*			*		**

Course Coordinator: Paolo Bucci

Department of Computer Science and Engineering
395 Dreese Laboratories
2015 Neil Avenue
Columbus, OH 43210-1277

CSE 221: Software Development Using Components

Description

Component-based software from client programmer’s perspective; intellectual foundations of software engineering; mathematical modeling; specification of object-oriented components; layering; testing and debugging layered operations.

Level, Credits, Class Time Distribution, Prerequisites

Level	Credits	Class Time Distribution	Prerequisites
U	4	3 cl, 1 1-hr lab	Math 151; 202 or 201 or En Graph 167 or CS&E Placement Level A (H221: Math 151 or equiv; 202 or 201 or En Graph 167 or CS&E Placement Level A; enrollment in honors program).

Quarters Offered

- Au, Wi, Sp, Su

General Information, Exclusions, Cross-listings, etc.

- RESOLVE/C++ is taught.
- H221 offered in Wi qtr.

Intended Learning Outcomes

- Master using basic C++ control structures and statements, the basic classes Boolean, Character, Integer, Real, Text, Character_IStream, Character_OStream, and related RESOLVE/C++ principles for clients, to write simple main programs and layered implementations of operations.
- Be familiar with using the computing environment (operating system, tools, language system, etc.) to complete lab assignments and to communicate electronically with others.
- Be familiar with using the Id_Name_Table, Least_Cost_Path_Machine, and Natural_Number classes to write simple main programs and layered implementations of operations.
- Be familiar with using simple recursion to write layered implementations of operations.
- Be familiar with using simple predicate calculus assertions involving mathematical integer, string, and tuple models to understand and reason about an operation’s behavior.
- Be familiar with using simple techniques to test layered implementations of operations, including developing and carrying out simple specification-based test plans.
- Be familiar with using simple techniques to debug layered implementations of operations.
- Be exposed to writing simple predicate calculus assertions involving mathematical integer, string, and tuple models to describe the intended behavior of an operation.
- Be exposed to using induction arguments to establish the correctness of recursive implementations of operations.
- Be exposed to using simple loop specifications to reason about loop behavior.
- Be exposed to using basic algorithm analysis techniques and notations to analyze and express execution time of operations whose implementations involve straight-line code and simple loops.

Texts and Other Course Materials

- *Software Component Engineering* - Bruce W. Weide
- *CSE 221 Handouts*
- *CSE 221: Software Development Using Components Getting Started* - Timothy J. Long

Topics

Number of Hours	Topic
4	Introductory material and computing environment
6	Component descriptions from client perspective
9	Specification and use of Id_Name_Table, Least_Cost_Path_Machine, Text, and Natural_Number components
3	Extending and checking components
2	Testing
2	Debugging
2	Performance analysis
3	Iteration
2	Recursion
3	Some fundamental algorithms such as binary search
4	Exams and review

Representative Lab Assignments

- Home page
- Newton Iteration
- Id_Name_Table client
- Least_Cost_Path_Machine client
- Text operations
- Recursive implementation of Text operations
- Natural Number calculator

Grades

Midterm exam	17%
Final exam	30%
Homework assignments	5%
Closed lab assignments	9%
Lab assignments	24%
In-class activities	15%

Relationship to BS-CSE Program Outcomes/EC 2000 Criterion 3 Outcomes

a	b	c	d	e	f	g	h	i	j	k
**	**	***	***	*	*	***	*	*		***

Course Coordinator: Timothy J Long

Department of Computer Science and Engineering
395 Dreese Laboratories
2015 Neil Avenue
Columbus, OH 43210-1277

CSE 222: Development of Software Components

Description

Templates for generalization and decoupling; container components; component-based software from implementer’s perspective; data representation using layering and using pointers.

Level, Credits, Class Time Distribution, Prerequisites

Level	Credits	Class Time Distribution	Prerequisites
U	4	3 1-hr lec, 1 1-hr lab	221 (H222: H221)

Quarters Offered

- Au, Wi, Sp, Su

General Information, Exclusions, Cross-listings, etc.

- RESOLVE/C++ is used.
- H222 offered in Sp qtr.

Intended Learning Outcomes

- Master using the computing environment (operating system, tools, language system, etc.) to complete lab assignments and to communicate electronically with others.
- Master using C++ class templates and classes, and related RESOLVE/C++ principles for clients, to write layered implementations of operations.
- Master using simple recursion to write layered implementations of operations.
- Master using simple techniques to test layered implementations of operations, including developing and carrying out simple specification-based test plans.
- Master using simple techniques to debug layered implementations of operations.
- Be familiar with using the Array, List, Partial Map, Queue, Record, Sequence, Set, and Stack templates to write layered implementations of operations.
- Be familiar with using basic C++ control structures and statements, RESOLVE/C++ class templates and classes, and related RESOLVE/C++ principles for clients and implementers, to write bodies of component realizations with layered data representations.
- Be familiar with reading code that uses dynamic storage management (new and delete) and pointers to create simple "raw C++" data representations.
- Be familiar with using loop invariants to reason about loop behavior.
- Be familiar with using basic algorithm analysis techniques and notations to analyze and express execution time of operations whose implementations involve straight-line code and simple loops.
- Be familiar with using simple formal logic assertions involving mathematical set models to understand and reason about an operation’s behavior.
- Be familiar with using simple techniques to test implementations of class templates that define new types, including developing and carrying out simple specification-based test plans.
- Be familiar with using simple techniques to debug implementations of class templates that define new types.
- Be exposed to using data representation conventions ("representation invariants") and correspondences ("abstraction relations") to reason about correctness of data representations.
- Be exposed to writing programs using dynamic storage management and pointers to create simple "raw C++" data representations.
- Be exposed to writing simple formal logic assertions involving mathematical set models to describe an operation’s behavior.
- Be exposed to using the RESOLVE/C++ principles for interface designers to guide the choice of

mathematical model and operation behavior of a new software component.

Texts and Other Course Materials

- *Software Component Engineering* - Weide, B.W., McGraw-Hill Custom Publishing, 2000.

Topics

Number of Hours	Topic
3	Templates; instantiates relation; component coupling diagrams and code; instantiation diagrams and code; Set and Sequence components
3	Checking components; checks relation using templates; Queue component
3	Extensions; extends relation using templates; toolkit classes; sorting Queues
3	Stack component; how recursion works; Record and Partial_Map components
3	Loop invariants
1	Performance analysis; analysis of algorithms
3	Representation; encapsulates relation; convention and correspondence
3	Array component; hashing to implement Partial_Map; fully parameterized components; uses relation; partial instantiation; specializes relation
6	Pointers and dynamic storage; List component
2	Review and exams

Representative Lab Assignments

- E-mail classifier
- Automated glossary generation
- Partial_Map represented using Array of Queue of Record (hashing)
- List represented using "raw C++" pointers

Grades

Midterm Exam	20%
Final Exam	30%
Homework Assignments	10%
Closed Lab Assignments	8%
Lab Assignments	30%
Class Participation	2%

Relationship to BS-CSE Program Outcomes/EC 2000 Criterion 3 Outcomes

a	b	c	d	e	f	g	h	i	j	k
***	*	***	**	**	*	**		*		***

Course Coordinator: Bruce Weide

Department of Computer Science and Engineering

395 Dreese Laboratories
2015 Neil Avenue
Columbus, OH 43210-1277

CSE 314: Business Programming with File Processing

Description

Business data processing principles and programming: sequential file processing algorithms, sorting, data validation. COBOL is taught.

Level, Credits, Class Time Distribution, Prerequisites

Level	Credits	Class Time Distribution	Prerequisites
U	4	3 cl	214

Quarters Offered

- Au, Wi, Sp

General Information, Exclusions, Cross-listings, etc.

Intended Learning Outcomes

- Master the following algorithms: single and multiple control breaks; matching, verification, and merge/purge; 1- and 2-dimensional tables.
- Master the designing and coding of well-structured COBOL programs and subprograms to process sequential files using any or all of the following program design tools: system flowcharts, hierarchy (structure) charts, flowcharts, pseudocode, print or screen charts.
- Be familiar with the COBOL reference card.
- Be familiar with debugging techniques including using the COBOL debugger.
- Be exposed to testing and data validation techniques.
- Be exposed to the definition, use and creation of makefiles.

Texts and Other Course Materials

- *COBOL for the 21st Century* - Stern, Stern & Ley

Topics

Number of Hours	Topic
3	Vocabulary; columns; margins; basic coding rules; typing in, compiling and running a COBOL program
3	Identification and environment division; data names; symbols; data division - file section; working storage; picture clauses; value clauses; group items; literals; constants; figurative constants; non-numeric literals; procedure division statements w/ file I/O (open, close, read, write); move statement (simple)
3	Perform statement (simple); putting it together; display; accept omitted; stop run; move statement rules; edited I/O
3	Compute and other arithmetic statements; accept time and date; scope terminators
3	Control structures; relational operators and relational expressions, sign and class tests; logical operators (AND,OR, NOT); implied conditions; condition names; evaluate statement

6	Single control break algorithm; string, unstring; double control break algorithm; debugging techniques; perform statement variations
3	Redefines; initialize; 2-dimensional tables; sort logic
6	Subprograms; sort/merge files; makefiles; testing and data validation techniques; additional sequential processing issues

Representative Lab Assignments

- Lab1 - Type in, compile and run a given COBOL program
- Lab2 - Simple report style COBOL program
- Lab3 - Arithmetic statements and Edited I/O
- Lab4 - Single Control Break
- Lab5 - Double Control Break
- Lab6 - Table (i.e., Array) processing
- Lab7 - Sort, Search, Merge and Makefiles

Grades

Participation	5%
Quizzes	10%
Midterm Exam	25%
Projects	30%
Final Exam	30%

Relationship to BS-CSE Program Outcomes/EC 2000 Criterion 3 Outcomes

a	b	c	d	e	f	g	h	i	j	k
*	*	***			*			*	*	*

Course Coordinator: Kathryn Reeves

Department of Computer Science and Engineering
395 Dreese Laboratories
2015 Neil Avenue
Columbus, OH 43210-1277

CSE 321: Case Studies in Component-Based Software

Description

Case studies using: tree and binary tree components and binary search trees; context-free grammars; tokenizing, parsing, and code generating components; sorting components and sorting algorithms.

Level, Credits, Class Time Distribution, Prerequisites

Level	Credits	Class Time Distribution	Prerequisites
U	4	3 1-hr lec, 1 1-hr lab	222/H222. Prereq or concur: Math 366

Quarters Offered

- Au, Wi, Sp, Su

General Information, Exclusions, Cross-listings, etc.

- RESOLVE/C++ is used.

Intended Learning Outcomes

- Master using the computing environment (operating system, tools, language system, etc.) to complete programming projects and to communicate electronically with others.
- Master using RESOLVE/C++ class templates and classes, and related RESOLVE/C++ principles for clients and implementers, to build new components with layered data representations.
- Master using simple techniques to debug implementations of class templates.
- Master using Binary_Tree template as a client.
- Be familiar with using layering of software components to implement a large programming project.
- Be familiar with using conventions ("representation invariants") and correspondences ("abstraction relations") to reason about layered data representations.
- Be familiar with using the RESOLVE/C++ principles for interface designers to guide the choice of mathematical model and operation behavior of a new software component.
- Be familiar with using simple techniques to test implementations of class templates including developing and carrying out simple specification-based and code-based test plans.
- Be familiar with using context-free grammars and languages to specify syntax.
- Be familiar with using recursive descent to process context-free languages.
- Be familiar with using state-transition diagrams to tokenize languages.
- Be familiar with using tokenizing software components and their implementations.
- Be familiar with implementing insert, delete, and search algorithms for binary search trees.
- Be familiar with using Tree template as a client.
- Be familiar with using Sorting_Machine template as a client and as an implementer.
- Be familiar with important classical sorting algorithms as implementations of Sorting_Machine template.
- Be familiar with using basic algorithm analysis techniques and notations to analyze and express execution time.
- Be familiar with working as part of a team to implement a large programming project.

Texts and Other Course Materials

- *Software Component Engineering, McGraw-Hill, 2000.* - Weide, B.W.
- *CSE 321 Course Notes, OSU Reprographics, 2005.*

Topics

Number of Hours	Topic
1	Course introduction
5	Binary_Tree component; Binary search trees; Partial_Map on BST
4	Project intro; Convention/Correspondence; Tree component
6	Statement and Program components
5	Context-free grammars; Recursive descent parsers
4	BL ("Bugs Language") program execution; Code generation
4	BL_Tokenizing_Machine component
6	Sorting and Sorting_Machine component
2	Performance analysis
3	Exams and review

Representative Lab Assignments

- Binary search tree implementation of Partial_Map
- Implementation of Statement and Pretty_Print
- Parser for Program/Statement
- Code generator for Program
- Implementation of BL_Tokenizing_Machine
- Heapsort implementation of Sorting_Machine

Grades

Midterm Exam	20%
Final Exam	30%
Homework Assignments	10%
Closed Lab Assignments	8%
Lab Assignments	30%
Class Activities/Participation	2%

Relationship to BS-CSE Program Outcomes/EC 2000 Criterion 3 Outcomes

a	b	c	d	e	f	g	h	i	j	k
**	*	***	***	**	*	**		*		***

Course Coordinator: Timothy J Long

Department of Computer Science and Engineering
395 Dreese Laboratories
2015 Neil Avenue
Columbus, OH 43210-1277

CSE 670: Introduction to Database Systems I

Description

Database systems use, logical database design, entity-relationship model, database normalization, query languages and SQL, relational algebra, and relational calculus; database design project.

Level, Credits, Class Time Distribution, Prerequisites

Level	Credits	Class Time Distribution	Prerequisites
UG	3	3 cl	314 or 222 or 230 or 502; Math 366

Quarters Offered

- Au, Wi, Sp

General Information, Exclusions, Cross-listings, etc.

Intended Learning Outcomes

- Master using relational databases.
- Master writing queries in relational data languages including SQL and relational algebra.
- Master using mechanisms for data independence, including data models, languages and views.
- Master logical database design.
- Be familiar with conceptual database design.

Texts and Other Course Materials

- *Fundamentals of Database Systems, Addison-Wesley, 4th edition* - Ramez Elmasri and Shamkant Navathe

Topics

Number of Hours	Topic
4	Introduction; Entity-Relationship (ER) Model
1	The Structure of the Relational Data Model
5	Relational Algebra and Relational Calculus
9	Functional Dependencies and Normalization
1	ER-to-Relational Data Model
5	SQL
1	Graphical User Interfaces
2	Embedded SQL
2	Review and Exam

Representative Lab Assignments

- Provide SQL code which declares, populates, and queries a database
- Define and use a database with MS Access

- Write a Java program with embedded SQL code

Grades

Homework	30%
Project	20%
Midterm Exam	20%
Final Exam	30%

Relationship to BS-CSE Program Outcomes/EC 2000 Criterion 3 Outcomes

a	b	c	d	e	f	g	h	i	j	k
***	*	**	***	**		*		*		***

Course Coordinator: Eitan Gurari

Department of Computer Science and Engineering
395 Dreese Laboratories
2015 Neil Avenue
Columbus, OH 43210-1277

Mathematics 366	**3 cr.**	**Discrete Mathematical**
A, W, Sp, Su (1st Term)		**Structures I**

Prerequisite:

Mathematics 132 or 152.

Catalog Description:

Mathematical formalization and reasoning, logic, Boolean algebra; sets, functions, relations, recursive definitions, mathematical induction; elementary counting techniques.

Purpose of Course:

To provide the foundation for a deeper understanding of the conceptual tools in computer science. Computers, however, are not used in this course. The desire of the CS&E faculty is that the course presents math in rigorous form and requires students to deal with abstract systems and mathematical proofs.

Follow-up Course:

Math 566.

Text:

Discrete Mathematics with Applications, S. S. Epp, 3rd edition

Topics List & Sample Syllabus:

Sections	Topics
	THE LOGIC OF COMPOUND SETS
1.1	Logical Form and Logical consequence
1.2	Conditional Statements
1.3	Valid and Invalid Arguments
1.4	Application: Digital Logic Circuits
	THE LOGIC OF QUANTIFIED STATEMENTS
2.1	Predicates and Quantified Statements I
2.2	Predicates and Quantified Statements II
2.3	Statements Containing Multiple Quantifiers
2.4	Arguments with Quantified Statements
	ELEMENTARY NUMBER THEORY AND METHODS OF PROOF
3.1	Direct Proof and Counterexample I: Introduction
3.2	Direct Proof and Counterexample II: Rational Numbers
3.3	Direct Proof and Counterexample III: Divisibility
3.4	Direct Proof and Counterexample IV: Division into Cases and the Quotient-Remainder Theorem
3.5	Direct Proof and Counterexample V: Floor and Ceiling
3.6	Indirect Argument: Contradicton and Contraposition
	SEQUENCES AND MATHEMATICAL INDUCTION
4.1	Sequences
4.2	Mathematical Induction I
4.3	Mathematical Induction II
4.4	Strong Mathematical Induction and the Well-Ordering Principle
	SET THEORY
5.1	Basic Definitions of Set Theory
5.2	Properties of Sets
5.3	Disproofs, Algebraic Proofs and Boolean Algebras
	RELATIONS
10.1	Relations on Sets
	FUNCTIONS
7.1	Functions Defined on General Sets
7.3	One-to-One and Onto, Inverse Functions
7.5	Composition of Functions

Course Coordinator:
Timothy Carlson
2006-2007