

CSC 7 Syllabus

Discrete Mathematics

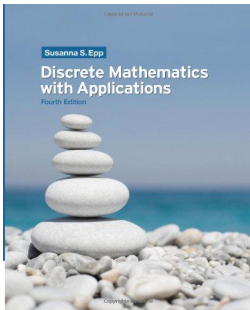
Instructor: [Dr. Mark E. Lehr](#)

E-Mail: mark.lehr@rcc.edu

Office Hours: [hyper_link](#)

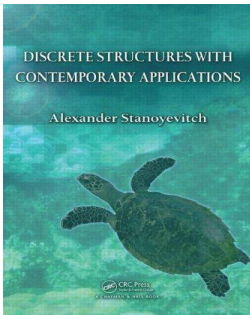
Outline: [hyper_link](#)

Required Text

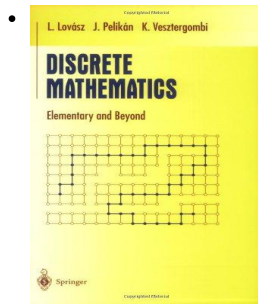


- Hardcover: 984 pages
- Publisher: Cengage Learning; 4 edition (August 4, 2010)
- ISBN-10: 0495391328
- ISBN-13: 978-0495391326

Optional Text



- Hardcover: 1002 pages
 - Publisher: Chapman and Hall/CRC; 1 edition (January 19, 2011)
 - ISBN-10: 1439817685
 - ISBN-13: 978-1439817681
-



Paperback: 284 pages

- Publisher: Springer; 2003 edition (February 17, 2003)
- Language: English
- ISBN-10: 0387955852
- ISBN-13: 978-0387955858

Required Materials

Computational device, i.e. a calculator will be useful. Multiple C++ compilers utilized from CSC 5. Additionally, you should bring your book to class each and every class meeting.

Course Description

This course is an introduction to the discrete mathematics/structures used in Computer Science with an

emphasis on their applications. Topics covered include: Functions, Relations and Set; Basic Logic; Proof Techniques; Basics of Counting; Graphs and Trees; and Discrete Probability. 54 hours lecture and 18 hours laboratory.

Prerequisite - CSC 5: Programming Concepts and Methodology I: C++

Before entering the course, students will be able to:

1. Summarize the evolution of programming languages illustrating how this history has led to the paradigms available today.

- CSC 5 - Describe the software development life-cycle
- CSC 5 - Explain what an algorithm is and its importance in computer programming.

2. Discuss the importance of algorithms in the problem-solving process. Identify the necessary properties of good algorithms.

- CSC 5 - Demonstrate different forms of binding, visibility, scoping, and lifetime management.
- CSC 5 - Describe the principles of structured programming and be able to design, implement and test structured programs.

3. Identify the information input requirements, synthesize the algorithmic steps needed to transform the data input into the required output information, and organize the output format to facilitate user communication.

- CSC 5 - Apply the principles of logical and programming concepts to develop solutions for gaming, business, scientific and mathematical problems.
- CSC 5 - Use pseudocode, flowcharts, and a programming language to implement, test, and debug algorithms for solving problems. Identify the information requirements, synthesize the algorithmic steps needed to transform the data input into the required output information, and organize the output format to facilitate user communication.

4. Create computer programs in C++ using the principles of structured programming. Analyze and explain the behavior of simple programs involving the fundamental programming constructs. Modify and expand programs that use standard conditional and iterative control structures and functions. Design,

implement, test, and debug programs that use fundamental programming constructs. Choose appropriate conditional constructs. Apply techniques of structured decomposition.

- CSC 5 - Create computer programs using the principles of structured programming and demonstrate the use of an IDE with appropriate libraries. Design, implement, test, and debug programs that use fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, and functions.

Course Objectives

Upon successful completion of the course, students should be able to:

1. Describe how formal tools of symbolic logic are used to model real-life situations, including those arising in computing contexts such as program correctness, database queries, and algorithms.
2. Relate the ideas of mathematical induction to recursion and recursively defined structures.
3. Analyze a problem to create relevant recurrence equations.
4. Demonstrate different traversal methods of trees and graphs.
5. Apply the binomial theorem to independent events and Bayes' theorem to dependent events.

Lab Hours

A total of **18** laboratory hours are **required to pass** this course. The first hour is to be recorded by the end of the second week of class. The 9th is to be completed prior to the 9th week of class. All 18 hours must be completed prior to the final. An F will be recorded for the class grade if less than 18 hours of lab are completed prior to the end of the semester.

School and Class Rules

Attendance and participation: If you do not attend you cannot participate; if you do not participate, you cannot learn and therefore will not pass this class. You must show up to class prepared and ready to learn. Daily attendance will be taken and has same requirements as lab.

Classroom decorum: Be on time to class. If the door is locked and class has started, you missed class for that day. No food or drink is allowed in the classroom. You are expected to be cooperative and respectful during class. Using your cell phone, disruptive talking or behavior is considered rude and you will be asked to leave if you insist on being rude; you will not receive points for that day. If you have an emergency and must use your cell phone, please step out of the classroom and return when you are done. You **do not** have the right to disrupt the learning of others.

Statement on Academic Dishonesty:

RCC defines plagiarism as, "Presenting another person's language (spoken or written), ideas, artistic works or thoughts as if they were one's own." This includes using someone else's C++ code. Plagiarism is academically dishonest. Students must make appropriate acknowledgment of the original source where material written or compiled by another is used." Cheating or dishonest practices, such as turning in the writing of someone else and claiming it as your own, will result in your receiving a failing grade on the assignment and possibly for the course.

ADA Information

Please let me know if you need accommodations for a documented disability. The office of Services to Students with Disabilities will also be able to provide help and assistance.

DIVERSITY STATEMENT

Riverside City College School of Business embraces a notion of an intellectual community enriched by diversity with multiple dimensions, including race, ethnicity and national origin, gender, gender identity, sexuality, class, and religion. We are particularly committed to populations that have historically been excluded from equitable participation in the classroom, higher education institutions, and our communities. Individually, we are devoted to addressing our unconscious bias to pave the way for a more inclusive curriculum and learning environment.

Course Activities and Class Format

Daily classroom instruction will consist of lectures, discussions, and demonstrations, as well as hands on work, both collaboratively and individually. Lecture will be delivered verbally, supported by PowerPoint presentations, chalk board drawings, and on occasion, paper handouts, among other methods. Periodically, students will be required to interact and work in groups or teams to reinforce learning.

Lab: Flowcharts/pseudocode/calculations/programs per assigned topic. Students may collaborate on solutions and coding. Peer talks will also be required discussing the lecture content.

Homework/Quizzes: Flowcharts/pseudocode/calculations/programs per assigned topic. Students may collaborate on homework solutions, but the code must be your own.

Mid-Term: No makeups. Calculations and Programs illustrating the concepts. No collaboration allowed.

Project I: A program of your choice to illustrate all the concepts covered up to the Midterm. Full documentation with a write-up including flowcharts/pseudocode. Your unique solution. Unused, or duplicative code, and unnecessary i/o will not count. Minimum lines of code and writeup will produce a C grade or less. This is one place to impress me with the knowledge and expertise you acquire in class.

Final Exam: No makeups. Calculations and Programs illustrating the concepts. No collaboration allowed.

Project II: A program of your choice to illustrate all the concepts covered from the Midterm to the Final. Full documentation with a write-up including flowcharts/pseudocode. Your unique solution. Unused, or duplicative code, and unnecessary i/o will not count. Minimum lines of code and writeup will produce a C grade or less. This is one place to impress me with the knowledge and expertise you acquire in class. This project will be presented on Finals day.

Reading Assignments: Reading assignments will be given and could be followed with impromptu quizzes which will be graded pass/fail and count as your attendance for lecture or lab.

Canvas

This course includes the use of Canvas. It is the student's responsibility to visit Canvas and stay current with respect to assignments and grades. Canvas will be used in the following ways:

Course materials: Course materials such as the syllabus, homework assignments, and review materials will be posted on Canvas. Reading assignments will be posted on Canvas as well.

Homework/Projects/Exams: will be posted from Canvas.

Announcements: In addition to making announcements in class, all announcements will be posted on Canvas.

Emails: All emails sent to and by me will be sent outside of Canvas using your student or personal email.

Your grade will be based on the following activities:

Activity	%	Point Value
Lab Assignments/Attendance	-6	25 points subtracted for each hour missed (2 sessions can be made up)
Class Lecture/Attendance	-2	10 points subtracted for each hour missed (2 sessions can be made up)
Homework	20	100 points total for all assignments
Project 1	20	100 points
Midterm	20	100 points
Project II	20	100 points
Final	20	100 points
Total Points		500

Grading

Your grade will be calculated as follows:

Grade	%	No. of points	Grade	%	No. of points
A	90 and above	450 and above	D	60 - 69.9	300-349
B	80 - 89.9	400-449	F	Below 60	299 and below
C	70 - 79.9	350-399			

Class Schedule (Subject to change)

COURSE CONTENT

Weeks 1-7

Basic Logic → Chapters 1 to 3 (1.1-1.3, 2.1-2.3, 3.1-3.4)

Proof Techniques → Chapters 4 and 5 (4.1-4.4, 4.6, 5.1-5.2, 5.6-5.7)

Week 8 → 4 Quizzes weekly prior to Midterm and Project I.

Weeks 9-15

Functions, Relations and Sets → Chapters 6 to 8 (6.1, 7.1-7.2, 8.1-8.3)

Basics of Counting and Discrete Probability → Chapter 9 (9.1-9.4)

Graphs and Trees → Chapter 10 and 11 (10.1, 10.5, 11.1-11.2)

Week 16 → 4 Quizzes weekly prior to Final and Project II.

Homework Assignments

Hmwk 1 Chapter 1

Put in 1 hour in the Lab this week before Census.

Section 1 → 1, 4, 7, 10

Section 2 → 1, 4, 7, 10

Section 3 → 1, 4, 7, 10, 13, 16, 19

Submit all homework in class, paper/pencil, deforestation is tolerated.

Hmwk 2 Chapter 2

All done by hand, single sided sheets of paper.

Section 1 → 6, 12, 18, 24, 30, 36, 42, 48

Section 2 → 6, 12, 18, 24, 30, 36, 42, 48

Section 3 → 6, 12, 18, 24, 30, 36, 42

Section 4 → 6, 12, 18, 24, 30

Section 5 → 6, 12, 18, 24, 30, 36, 42

Hmwk 3 Chapter 3

Attached Files:

[CSC7Prob3p4p32.JPG](#) (414.66 KB)

Type your answers on the front pages but attach your work after the answer pages.

Section 1 → 4, 8, 12, 16, 20, 24, 28, 32

Section 2 → 6, 12, 18, 24, 30, 36, 42, 48

Section 3 → 10, 20, 30, 40, 50, 60

Section 4 -> 4, 8, 12, 16, 20, 24, 28, 32

Hmwk 4 Chapter 4

Read Chapter 4

Section 1 -> 4,5,11,12,15,20,25,31,38,43

Section 2 -> 4,7,10,14,15,20,24,27,37,44

Section 3 -> 4,9,10,13,15,22,23,27,32,40,44

Section 4 -> 5,8,11,16,20,22,26,31,38,44

Section 6 -> 4,5,10,12,15,21,22,27,31,34

Hmwk 5 Chapter 5

Turn in all work in Class

Section 1 -> 1,8,18,27,35,44,51,53,61,75,88

Section 2 -> 3,8,14,15,20,23,28,30,33,37

Section 6 -> 3,7,13,17,20,24,27,33,39,41

Section 7 -> 5,10,14,19,25,30,35,39,45,50

Hmwk 6 Chapters 6,7,8

Submit the previous hmwk and this one in class.

Section 6.1 - 5,8,9,13,16,21,24,28,29,32

Section 7.1 - 6,10,15,18,25,30,35,40,45,51

Section 8.1 - 2,4,5,8,10,12,16,19,23,24

Section 8.2 - 6,11,15,20,25,31,37,43,51

Section 8.3 - 5,10,11,15,20,26,32,38,42

Hmwk 7 Chapter 9

Submit the previous hmwks uncollected and this one in class with the quiz.

Section 9.1 - 3,7,9,12,16,17,21,24,28,32

Section 9.2 - 4,11,16,20,24,31,35,41,45

Section 9.3 - 4,11,15,20,25,26,31,35,40,43

Section 9.4 - 5,9,12,17,20,23,29,32,36,38

Hmwk 8 Chapter 9,10,11 06/14/15

Five problems from each section

Section 9.7 5,10,15,25,31

Section 9.8 8,16,24,32,40

Section 9.9 5,10,15,25,31

Section 10.1 8,16,24,32,40

Section 10.5 5,10,15,25,31

Section 11.1 5,10,25,20,25

Section 11.2 10,20,31,40,50

Lab Assignments

Lab 1 Luhn Algorithm

Attached Files:

[GeneratingLuhnNumber.zip](#) (10.077 KB)

Read Chapter 1 from the Text, and code

If you don't know where to begin, use my program as a starting point.

Credit Card error detection check -> Luhn Algorithm, (<http://www.freeformatter.com/credit-card-number-generator-validator.html#cardFormats>)

From the rightmost digit, which is the check digit, moving left, double the value of every second digit; if the product of this doubling operation is greater than 9 (e.g., $7 * 2 = 14$), then sum the digits of the products (e.g., 10: $1 + 0 = 1$, 14: $1 + 4 = 5$).

Take the sum of all the digits. If the total modulo 10 is equal to 0 (if the total ends in zero) then the number is valid according to the Luhn formula; else it is not valid.

Assume an example of an account number "7992739871" that will have a check digit added, making it of the form 7992739871x:

Account number	7	9	9	2	7	3	9	8	7
Double every other	7	18	9	4	7	6	9	16	7
Sum of digits	7	9	9	4	7	6	9	7	7

The check digit (x) is obtained by computing the sum of digits then computing 9 times that value modulo 10 (in equation form, $(67 * 9 \bmod 10)$). In algorithm form:

Compute the sum of the digits (67).

Multiply by 9 (603).

The last digit, 3, is the check digit.

Submit here and copy yourself on an email to mark.lehr@rcc.edu with subject: Lastname, Firstname - Lab 1 Luhn Algorithm – Section Code

Lab 2 Luhn Algorithm

You will use the concepts from your previous program and the reference to create your next program.

1) Create an enumerated data type called CrdCard

The types of Credit Cards to create are American Express, Visa, MasterCard, Discover, ALL

The ALL just means could be any of the 4 specified.

2) Create a function that takes in the enumerated type and returns a valid credit card number using the

Luhn Algorithm. Note: Length, etc... are dependent on credit card type.

genCC(CrdCard, char []), or

char *genCC(CrdCard)

3) Create a function that randomly flips only 1 digit.

void flipDig(char []);

4) Create a function that determines if the result is a valid Credit Card

bool validCC(char[]);

5) Loop 10,000 times and record how many valid vs. invalid Credit Cards are detected.

Note: You are going to have to test out your generate credit card function extensively. The length is not always fixed even for a given credit card.

Submit here, copy yourself on an email to mark.lehr@rcc.edu with subject: Lastname, Firstname - Lab 2 Luhn Algorithm – Section Code

<http://www.getcreditcardnumbers.com/>

Lab 3 Luhn Algorithm

You will use the concepts from your previous program and the reference to create your next program. Very similar to last assignment.

For each,

1) transpose 2 numbers,

2) flip 2 numbers

Loop 10,000 times and record how many valid vs. invalid Credit Cards are detected for 1) and 2)

Submit here, copy yourself on an email to mark.lehr@rcc.edu with subject: Lastname, Firstname - Lab 3 Luhn Algorithm – Section Code

Lab 4 Hashing

Refer to the following website

<http://www.partow.net/programming/hashfunctions/>

Download the code for c/c++ hashing functions.

Get all the hashes working with the following string in Quotes

Horatius from the Lays of Ancient Rome XXVII

"Then out spake brave Horatius,

The Captain of the Gate:
To every man upon this earth
Death cometh soon or late.
And how can man die better
Than facing fearful odds,
For the ashes of his fathers,
And the temples of his gods."

Submit here, copy yourself on an email to mark.lehr@rcc.edu
with subject: Lastname, Firstname - Lab 4 Hashing – Section Code

Lab 5 Bloom Filter

Attached Files:

[BloomFilterDerivation.JPG](#) (269.226 KB)

[BloomFilter.ods](#) (22.861 KB)

[BloomFilter.pdf](#) (35.526 KB)

Refer to this example

<http://billmill.org/bloomfilter-tutorial/>

- 1) Using the 2 of the 9 hash functions you have, duplicate the process found in the above example.
- 2) Next create a file with random inputs and use all 9 hashes this time
<http://pages.cs.wisc.edu/~cao/papers/summary-cache/node8.html>
- 3) Choose several spots in the table with 9 hashes and confirm results with your program.

False Positive $\rightarrow (1 - e^{-(kn/m)})^k$ where

m is the bit vector size

k are the number of hashes

n are the number of keys used in the hash to fill the vector.

Lab 6 Hashcash

Attached Files:

[SHA1-Hashcash.zip](#) (73.557 KB)

Refer/read the following references/websites

<http://en.wikipedia.org/wiki/Hashcash>

<http://www.hashcash.org/>

<https://en.bitcoin.it/wiki/Hashcash>

See attached C++ and html programs

Utilize the 9 hash functions from the Lab 4 Assignment to perform a similar examination of deriving your own hashcash/proof of work test function. The difference is, I used SHA1 which is a hash function that returns a 40 hex character or 20 byte representation of a hash. You are using the 9 hash functions from the last assignment which only returns a 4 byte unsigned integer. That being the case, you can't look for '0' characters in front of your hash, but you can look at ranges. So take statistics on a hash $<10^9$, $<10^8$, $<10^7$, etc.....

Submit here, copy yourself on an email to mark.lehr@rcc.edu

with subject: Lastname, Firstname - Lab 6 Hashcash – Section Code

Lab 7 Merkle Tree 06/08/15

Attached Files:

[MerkelTree.png](#) (6.465 KB)

Given the attached Merkle Tree from http://en.wikipedia.org/wiki/Merkle_tree

Let

L1=Then out spake brave Horatius,

The Captain of the Gate:

L2="To every man upon this earth

Death cometh soon or late.

L3=And how can man die better

Than facing fearful odds,

and

L4=For the ashes of his fathers,

And the temples of his Gods."

Create all hashes 0-0, 0-1, 1-0, 1-1, 0, 1 with top

Show all you need to do to confirm

L1 is 0-1, 0, top

L2 is 0-0, 1, top

L3 is 1-1, 0, top and

L4 is 1-0, 0, top

Submit here and copy yourself on an email to mark.lehr@rcc.edu with subject: Lastname, Firstname - Merkle Tree – Section Code