***Discrete Structures***

|  |  |
| --- | --- |
| **COURSE No.:** | **???** |
| **PROGRAM:** | BITL, BBA (elective) |
| **INSTRUCTOR:** | Kalvis Apsītis |
| **CLASS DAYS & TIME:** | **Spring 2020 Semester:** Tuesdays, Thursdays (and Fridays) January 6 till April 24, 2020. |
| **OFFICE LOCATION & HOURS:** | near Room 417 |
| **CONTACT PHONE:** | **+371 29112997** |
| **E-MAIL:** | kalvis.apsitis@rbs.lv |

**TEXTBOOK**

**The Textbook:** Kenneth H.Rosen. Discrete Mathematics and Its Applications, 8th Edition. McGraw-Hill Education. 2019.

**Class Website:** <http://linen-tracer-682.appspot.com/discrete/index.html> (hosting may change – the permanent URL will be announced to the students).

**COURSE OBJECTIVES**

The course objectives are:

* To communicate with humans and computers accurately using mathematical structures, proof techniques and data structures.
* To prepare students for advanced programming techniques and algorithms used in computer networks, data encryption, financial transactions, Internet search and social media.
* To experiment with methods of creative thinking, solving problems in number theory, combinatorics, descriptive statistics and other IT-related fields of human knowledge.

**COURSE OVERVIEW**

Discrete Structures (also named Discrete Mathematics) serves as the introduction of formal logic and its illustrations that has been exceptionally fruitful to maintain technological progress and and modern computing. Formal logic has long history, it has been used by philosophers, lawyers, mathematicians and scientists since ancient times. Up to the 19th century the main vehicle to learn these techniques was Geometry (including the study of a classical text “Euclid’s Elements”).

Since 1940-ies electronic computers have revolutionized our ability to use and to automate logic and proof techniques with increasingly powerful computing devices. The practical interest has significantly shifted from geometry to other practically important fields of formal knowledge: set theory, Boolean logic and quantifiers, relational algebra and databases, the study of large integers and their arithmetic (also used in cryptography, block-chains and cryptocurrencies), graphs and combinatorics that describe our computing infrastructure.

In order to learn the techniques of Discrete Mathematics we will use these methods and tools:

(1) **Class Participation:** Attending class sessions, taking part in polls and other activities, presenting a proof of Discrete Mathematics in the classroom. The goal of these activities is to avoid unnecessary stress and to feel oneself at ease when you are doing mathematics.

(2) **Automated tests:** Some computer-graded questions are effective to verify your understanding of key concepts and procedures. There are important limitations - it is not possible to test every kind of mathematical knowledge with multiple-choice tests and other CBT (computer-based training) techniques. Please refer to the Essay by Terrence Tao. [On Multiple-Choice Exams in Mathematics, 2008-10-14](https://terrytao.wordpress.com/2008/12/14/on-multiple-choice-questions-in-mathematics/). The goal of automated tests is to exercise your intuition and the ability to apply the math knowledge to some tasks that only need a “back of the envelope” calculation.

(3) **Solving mathematical problems in writing** develops your ability to explain logical reasoning in English. This will be the main activity in homeworks and also in both exams - Midterm and Final. Problem solving is worth 500 points (so it is 50% of your total grade).

(4) **Seminar-like discussions** in small teams sometimes precede automated tests and exams in the class. During these discussions students will tell their team about some math result that will be tested shortly. The goal of this activity is to encourage discussion, speaking and listening skills and to cover some supplementary material from the course book. The participants will read the proofs before the class, so all the material for the seminars will be announced in advance.

(5) **Coq Proof Assistant and Computations:** Some Discrete mathematics are best done on computers. Proof checking on computers (Coq) provides a fresh perspective when compared with human-readable proofs that typically have considerable omissions. Computations (using Python, Scala or some other basic computation environments) may involve large prime numbers, experiments in cryptography, and combinatorial analysis. The goal of these activities is to show some applications of discrete mathematics that may be too time-consuming to do by hand. (The practicing of computational techniques and tools is only secondary in this case.)

**COURSE REQUIREMENTS**

In order to get a satisfactory grade, there are some mandatory activities that you need to complete.

Class Participation: Students will be graded on their attendance and contribution to the class. Therefore, it is important that you attend classes regularly. Anyone that misses two classes needs to visit the instructor during Office Hours (once per every 2 missed classes) to assess the knowledge gaps and possible remediation steps.  
Written assignments (submitted files): All assignments are due at the beginning of class on the due date. These will include several sets of math problems - mostly proofs or applying your theory knowledge in specific contexts.

Make-up exams: Make-up exams for the mid-term and the final are generally not given. If there are extenuating circumstances and you must miss an exam the instructor must be notified ahead of time. The only time a make-up exam can be taken is during the week following the date of the exam. If a student does not notify the instructor of an absence or misses the make-up exam deadline, the exam will not be included in the final grade.

**GRADING**

Grading for the course is as follows:

Final grades are calculated on the following basis.

> 949 = 10

850-949 = 9

750-849 = 8

650-749 = 7

550-649 = 6

450-549 = 5

350-449 = 4

< 350 = failing

Your grades will be updated regularly on the online assistant. Please check to see that your grades are recorded correctly.

*Assignments*  Points

|  |  |
| --- | --- |
| Class Participation and a Presentation | 100 |
| Automated Tests | 200 |
| Coq Proof Assistant and Computations | 200 |
| Homeworks | 200 |
| 2 Written Exams (Midterm and Final) | 300 |
| TOTAL | 1000 |

**ACADEMIC INTEGRITY**

To strengthen ethics within Riga Business School and the business community the RBS policy is to take steps to avoid cases of academic fraud. Be aware that any student who turns in written work that is not original with incorrectly referenced sources (i.e. plagiarized) will be subject to the RBS sanctions policy on Academic Fraud (see the policy on the online assistant). The same consequences apply to academic dishonesty on tests and quizzes.

To find information on what plagiarism is and how to avoid it please visit the links at:  
<http://www.uottawa.ca/plagiarism.pdf>  
<http://www.socialsciences.uottawa.ca/pdf/plagiarism2.pdf>  
<http://www.sass.uottawa.ca/writing/plagiarism.pdf>

This short presentation on research and plagiarism will also help learn to correctly reference sources and provides good advice on research:  
<http://library.acadiau.ca/tutorials/plagiarism/>

**CLASS SCHEDULE**

A specific schedule of assignments is listed below. You are responsible for having assigned readings and assignments read before class.

| Class | Topic | Pre-class preparation | Deliverables and Activities |
| --- | --- | --- | --- |
| Week1 | Propositional logic, proofs, predicates and quantifiers | Read Chapter 1 |  |
| Week 2 | Theory of sets, functions, relations, cardinality (and different sizes of infinity) | Read Chapter 2 |  |
| Week 3 | The theory of algorithms, unsolvable problems, complexity of algorithms | Read Chapter 3 |  |
| Week 4 | Number theory | Read Chapter 4 |  |
| Week 5 | Mathematical induction, recursion, invariants | Read Chapter 5 |  |
| Week 6 | Counting | Read Chapter 6 |  |
| Week 7 | Discrete Probability | Read Chapter 7 |  |
| Week 8 | More Counting Techniques | Read Chapter 8 |  |
| Week 9 | Relations. | Read Chapter 9 |  |
| Week 10 | Graphs. | Read Chapter 10 |  |
| Week 11 | Trees. | Read Chapter 11 |  |
| Week 12 | Boolean functions. | Read Chapter 12 |  |
| Week 13 | Levels of computability. | Read Chapter 13 |  |