MODULE SPECIFICATION

Part 1: Information			
Module Title	Discrete Mathematics [TSI]		
Module Code	UFCFRW-12-1	Level	Level 4
For implementation from	2021-22		
UWE Credit Rating	12	ECTS Credit Rating	6
Faculty	Faculty of Environment & Technology	Field	
Department	FET Dept of Computer Sci & Creative Tech		
Module Type:	Standard		
Pre-requisites	None		
Excluded Combinations	None		
Co-requisites	None		
Module Entry Requirements	None		
PSRB Requirements	None		

Part 2: Description

Educational Aims: The aim of this module is to instruct students in methods and models of computer mathematics: logical functions, logical networks, finite automaton, graph theory, flows in networks, algorithm theory, formal systems (theories). The module adopts the use of modern software such as R, Python, Mathcad to complete the practical assignments. The module provides a strong practical element giving ample opportunity to learn and practise new skills.

Outline Syllabus: Introduction to graph theory;
Non-oriented graph main characteristics. Eulerian graphs;
Trees and oriented trees. Spanning trees of the non-oriented graphs. The Kruskal's algorithm of minimal spanning tree searching;
Tree of shortest paths and its searching algorithm;
Networks and flows in the networks. Algorithm of the maximal flow and the minimal cost flow searching;
Transport problem;
Definition and representation ways of logical functions (LF). Elementary (LF) and their properties. LF superposition;
Disjunctive and conjunctive normal forms of LF. Completeness of LF family. Post's theorem;
A problem of LF minimization. Quine - Mc Klosky algorithm for the enumeration of LF simple

implicative functions. LF minimal form searching. Covering problem;
Finite state automatons (machines);
A minimization of the numbers of states for the finite automaton;
Intuitive requirements to algorithm. Various models of the algorithms. Church thesis;
A definition and representation ways of the Turing machine, Universal Turing;
machine. The way of its construction;
Stopping and self-applicability problem;
A proof of its algorithmically non-resolvability. General algorithm theory;
Introduction to the logic. Predicates and propositions, Truth formulas;
Principles of formal systems (theories) construction, Propositional calculus;
Logic of the propositional calculus;
Predicate calculus. Logic of predicate calculus;
The formal arithmetic;
Meta-theory of the formal systems;

Teaching and Learning Methods: This module will adopt a mix of lectures and practical classes. During lectures, theoretical aspects of the course will be provided to students by the teaching staff. Lectures will be supported by presentation published and available to the students on e.tsi.lv under the module section. Also, additional materials, like publications on the internet, videos etc will be presented in e.tsi.lv.

During practical classes each student receives an individual task to perform. Each practical task should be completed and uploaded to e.tsi.lv (under specific practical task element), it will be checked by the teaching staff and feedback will be provided. Defense of the performed task is not required. The teaching staff note the task is passed or failed. If the student has not completed the tasks, then they are not allowed to pass the exam. Modern software such as R, Python or Mathcad will be used in practical classes (students' choice). As a demonstration of the idea of the algorithm studied in the current practical task, the code on the Mathcad is provided as the most formalised and similar to pseudocode.

Part 3: Assessment

This module assessment is split into two components (A – Exam, B – Practical Assignments):
A - final 3-hour examination which will assess the students understanding of taught material that forms part of the learning outcomes but cannot easily be assessed through practical tasks.

B – component will consist from multiple elements, which are practical assignments. Each assignment should be completed, and written report should be provided to the teaching staff.

Element B1: A series of practical assignments, which provides results of assignments execution. This element is worth

Element B2 – a midterm in-class written closed-book tests covering graph theory, logical functions, finite automatons and Turing machine.

First Sit Components	**Final Assessment**	**Element weighting**	**Description**
Examination - Component A	✓	60 %	Examination
Portfolio - Component B		4 %	A portfolio of 4 practical assignments, which provides results of assignments execution.
In-class test - Component B		36 %	midterm in-class written closed-book test on graph theory. and midterm in-class written closed-book test on logical functions, finite automatons and Turing machine.

Resit Components	Final Assessment	Element weighting	Description
Examination - Component A		60 %	Examination
Portfolio - Component B		4 %	A portfolio of practical assignments, which provides results of assignments execution.
In-class test - Component B		36 %	written closed-book test on graph theory. and written closed-book test on logical functions, finite automatons and Turing machine.

Part 4: Teaching and Learning Methods

Learning Outcomes

On successful completion of this module students will achieve the following learning outcomes:

Module Learning Outcomes	Reference
Apply computational mathematical models and methods for solving practical problems	MO1
Specify graphs in different ways	MO2
Solve various problems in graph theory, for example, the problem of finding the shortest path, the problem of finding the maximum flow, and so on	MO3
Solve various problems in the theory of logical functions	MO4
Build a finite state machine and implement any algorithm using a Turing machine	MO5
Solve various problems in the theory of algorithms and formal systems	MO6
Use different modern software such as Mathcad, R or Python	MO7

Contact Hours

Independent Study Hours:	
Independent study/self-guided study	96
Total Independent Study Hours:	96
Scheduled Learning and Teaching Hours:	
Face-to-face learning	64
Total Scheduled Learning and Teaching Hours:	64
Hours to be allocated	120
Allocated Hours	160

Reading List

The reading list for this module can be accessed via the following link:

https://rl.talis.com/3/uwe/lists/3E565164-0661-0B67-D58F-7F3A4F8C1019.html?lang=en-gb&login=1

Part 5: Contributes Towards
This module contributes towards the following programmes of study: Computer Science and Software Development [Oct][FT][TSI][4yrs] BSc (Hons) 2020-21 Computer Science and Software Development [Oct][PT][TSI][5yrs] BSc (Hons) 2020-21 BSc (Hons) 2020-21 Computer Science and Software Development [Feb][FT][TSI][4yrs] BSc (Hons) 2020-21 Computer Science and Software Development [Feb][PT][TSI][5yrs] BSc (Hons) 2020-21 BSc (Hons) 2020-21