MODULE SPECIFICATION

Part 1: Information			
Module Title	Optimisation Methods [TSI]		
Module Code	UFCFTW-6-1	Level	Level 4
For implementation from	2021-22		
UWE Credit Rating	6	ECTS Credit Rating	3
Faculty	Faculty of Environment & Technology	Field	Computer Science and Creative Technologies
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 the course is to provide theoretical foundations of analytical optimisation techniques and prepare students for formulation and solution of applied optimisation problems. The set of discussed optimisation techniques include unconstrained analytical optimisation of univariate and multivariate functions, constrained optimisation using Lagrange multipliers, and linear programming using the simplex algorithm.

Outline Syllabus: Optimisation problem. Classification of optimisation problems and methods;
Unconstrained univariate optimisation. Theorems about necessary and sufficient conditions of a local extremum;
Unconstrained multivariate optimisation. Definition of a gradient, differential, Hessian. Theorems about necessary and sufficient conditions of a multivariate function's local extremum;
Constrained optimisation with equality constraints. Optimisation techniques: replacement, constrained variation, a method of Lagrange multipliers. Interpretation of Lagrange multipliers;
Constrained optimisation with inequality constraints. Introduction of slack variables and their meaning. Active and inactive conditions, Karush-Kuhn-Tucker conditions;
Convex programming. Convex sets and functions. Strong and weak Slater conditions. Kuhn-Tucker theorem for a convex programming problem;

Linear programming: problem formulation and real-world examples;
Linear programming: key theorems and geometrical solution of linear programming;
Linear programming: Simplex algorithm, Dantzig's theorems.

Teaching and Learning Methods: Learning and teaching will be provided to students in three forms: lectures, practical classes, and computer labs. 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 optimisation calculus classes, students receive a set of practical problems for every topic to perform. Several practical problems are explained during classes by a teaching assistant, while other problems are provided as a homework.

Computer labs are devoted to programming of optimisation algorithms using any software such as MathCad. Requirements' description for computer labs are provided, and students are expected to conduct the job independently outside the classes. The classes are reserved for requirement clarifications, problem discussion, and assessment.

Part 3: Assessment

This module assessment is split into two components (A – Exam, B – In-semester Assignments):

Both components should be completed individually (i.e. this is not group work).

A – the exam component that includes both theoretical and practical parts and consists of 2 elements:

Element A1 – a set of practical optimisation problems, which are expected to be solved within 2 hours. A1 represents 30% of the final module grade.

Element A2 – a written 2-hour closed-book exam of theoretical questions, randomly selected from a predefined list. Students are expected to demonstrate their knowledge of most important terms and algorithms as well as understanding of general concepts of optimisation. A2 represents 30% of the final module grade.

B – the component that includes all in-semester assignments.

Element B1– a portfolio covering a set of problems on every topic including optimisation calculus. (Source codes and supplementary reports should be provided).

First Sit Components	**Final Assessment**	**Element weighting**	**Description**
Practical Skills Assessment - Component A		30 %	optimisation calculus (2-hours)
Examination - Component A	✓	30 %	Theoretical examination (2 hours)
Portfolio - Component B		40 %	A series of set problems covering Optimisation calculus, followed by written lab reports
Resit Components	**Final Assessment**	**Element weighting**	**Description**
Practical Skills Assessment - Component A		30 %	Optimisation calculus (2-hours)
Examination - Component A		30 %	Theoretical examination (2 hours)
Written Assignment - Component B		40 %	A series of set problems covering Optimisation calculus, followed by written lab reports

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
Identify main types of optimisation problems and corresponding optimisation techniques	MO1
Articulate purposes and shortcomings of modern optimisation methods	MO2
Demonstrate knowledge of unconstrained and constrained non-linear univariate and multivariate optimisation	MO3
Solve optimisation problems with the Lagrange multipliers method	MO4
Solve linear programming problems with the simplex method	MO5
Identify optimisation problems in different applied spheres and formulate them in a proper formal way	MO6
Classify an applied optimisation problem and select an appropriate optimisation method for its solution	MO7
Execute post-optimisation analysis of an applied problem	MO8

Contact Hours

Independent Study Hours:	
Independent study/self-guided study	48
Total Independent Study Hours:	48
Scheduled Learning and Teaching Hours:	
Face-to-face learning	32
Total Scheduled Learning and Teaching Hours:	32
Hours to be allocated	60
Allocated Hours	80

Reading List

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

https://rl.talis.com/3/uwe/lists/94779640-953F-F756-F9A5-4936CF6931AE.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 [Feb][FT][TSI][4yrs] BSc (Hons) 2020-21