



## **Module Specification**

### **Compiler Construction. [TSI]**

Version: 2023-24, v2.0, 17 Mar 2023

#### **Contents**

<b>Module Specification .....</b>	<b>1</b>
<b>Part 1: Information .....</b>	<b>2</b>
<b>Part 2: Description .....</b>	<b>2</b>
<b>Part 3: Teaching and learning methods .....</b>	<b>3</b>
<b>Part 4: Assessment.....</b>	<b>4</b>
<b>Part 5: Contributes towards .....</b>	<b>5</b>

## Part 1: Information

**Module title:** Compiler Construction. [TSI]

**Module code:** UFCFHX-12-3

**Level:** Level 6

**For implementation from:** 2023-24

**UWE credit rating:** 12

**ECTS credit rating:** 6

**Faculty:** Faculty of Environment & Technology

**Department:** FET Dept of Computer Sci & Creative Tech

**Partner institutions:** Transport and Telecommunication Institute

**Delivery locations:** Not in use for Modules

**Field:** Computer Science and Creative Technologies

**Module type:** Module

**Pre-requisites:** None

**Excluded combinations:** None

**Co-requisites:** None

**Continuing professional development:** No

**Professional, statutory or regulatory body requirements:** None

## Part 2: Description

**Overview:** Not applicable

**Features:** Not applicable

**Educational aims:** The aim of this module is to give the students' knowledge of the main principles of compilers structure and programming as well as of the main

methods and principles for building different phases of compiler – lexical and syntax analyzer, internal form of program representation, optimization and code generation.

**Outline syllabus:** Introduction in automation of programming;

Formal systems;

The lexical analysis and programming of the lexical analyser (scanner);

Parsing and programming of a parser;

The internal form of representation of the initial program and semantics of tables;

Procedures of the semantic analysis;

Machine-independent optimisation;

Generation of codes;

Machine-sensitive optimisation;

Distribution of memory;

The target information of the compiler.

### **Part 3: Teaching and learning methods**

**Teaching and learning methods:** Learning and teaching will be provided to students in two forms: lectures and 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 code examples, text books, publications on the internet, videos etc will be presented in e.tsi.lv.

During practical tasks, each student receives an individual task to perform.

In addition to learning activities during taught sessions, students are expected to spend time outside of class on independent learning activities. These might include completing assignment tasks, independent reading, practising new skills on personal projects and watching informative videos, completing self-assessment test etc.

**Module Learning outcomes:** On successful completion of this module students will achieve the following learning outcomes.

**MO1** Know the main principles of construction of the translating programs

**MO2** Use principles of formal systems for the description of various designs of the programming language

**MO3** Carry out decomposition of the text of the source program (to build grammar of lexemes, final automata devices and their states diagrams, to build trees of syntactic analysis)

**Hours to be allocated:** 120

**Contact hours:**

Independent study/self-guided study = 96 hours

Face-to-face learning = 64 hours

Total = 160

**Reading list:** The reading list for this module can be accessed at [readinglists.uwe.ac.uk](https://rl.talis.com/3/uwe/lists/4132CB37-70D8-0FA0-5A46-AB9EAB1B2C42.html?lang=en-gb&login=1) via the following link <https://rl.talis.com/3/uwe/lists/4132CB37-70D8-0FA0-5A46-AB9EAB1B2C42.html?lang=en-gb&login=1>

## Part 4: Assessment

**Assessment strategy:** This module assessment is split into two:

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.

A series of labs, connected with lexical and syntax analysers (parser) development using any high-level programming language. Each lab as an application and its source code should be completed and uploaded to e.tsi.lv (under specific practical task element) in form of the report,

The assessment includes demonstration of the developed applications. The defence is happening orally and consists of discussion on theoretical issues which fits current lab.

**Assessment components:**

**Examination (First Sit)**

Description: Written Examination

Weighting: 60 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO2, MO3

**Portfolio (First Sit)**

Description: A series of labs, connected with lexical and syntactical analysers (parser) development using any high-level programming language

Weighting: 40 %

Final assessment: No

Group work: No

Learning outcomes tested: MO1, MO2, MO3

**Examination (Resit)**

Description: Written Examination

Weighting: 60 %

Final assessment: No

Group work: No

Learning outcomes tested: MO1, MO2, MO3

**Portfolio (Resit)**

Description: A series of labs, connected with lexical and syntactical analysers (parser) development using any high-level programming language

Weighting: 40 %

Final assessment: No

Group work: No

Learning outcomes tested: MO1, MO2, MO3

**Part 5: Contributes towards**

This module contributes towards the following programmes of study:

Computer Science and Software Development {Double Degree} [Oct][FT][TSI][4yrs]

BSc (Hons) 2020-21

Computer Science and Software Development {Double Degree} [Oct][PT][TSI][5yrs]

BSc (Hons) 2020-21

Computer Science and Software Development {Double Degree} [Feb][FT][TSI][4yrs]

BSc (Hons) 2020-21

Computer Science and Software Development {Double Degree} [Feb][PT][TSI][5yrs]

BSc (Hons) 2020-21