Module Specification

Functional Programming [TSI]

Version: 2023-24, v2.0, 07 Aug 2023

Contents

Module Specification 1

Part 1: Information 2

Part 2: Description 2

Part 3: Teaching and learning methods 3

Part 4: Assessment....4

Part 5: Contributes towards 5

Part 1: Information

Module title: Functional Programming [TSI]

Module code: UFCE84-6-1

Level: Level 4

For implementation from: 2023-24

UWE credit rating: 6

ECTS credit rating: 3

College: College of Arts, Technology and Environment

School: CATE School of Computing and Creative Technologies

Partner institutions: Transport and Telecommunication Institute

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: Discover the elements of the functional programming style. Learn the most powerful features and how to apply them purposeful in your daily programming tasks. Our team will guide you how to use Functional Programming as a dynamic tool to solve many real-world problems by writing robust, testable, and bug-free code.

Features: Not applicable

Educational aims: Introduce fundamentals of functional programming concepts, differentiating from the rest of programming paradigms and provide students with skills in software engineering and approached to problem solving.

Outline syllabus: Introduction to functional programming paradigm;
Functional language elements and data types (ie HASKELL);
Recursion;
Program execution;
Lambda-calculus;
Imperative program analogies;
LISP Programming Language.

Part 3: Teaching and learning methods

Teaching and learning methods: Learning and teaching will be provided to students in forms of lectures, labs, 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 code examples, text books, publications on the internet, official documentation, videos etc will be presented in e.tsi.lv.
During labs, each student receives an individual task to perform.

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

MO1 Understand and apply the concepts of functional programming and lambda-calculus.

MO2 Solve common programming challenges, adeptly identify the suitable data types for different scenarios, and skilfully comprehend and analyse programs created by others.

MO3 Effectively break down provided problems into manageable components and then construct functional programs using appropriate function definitions, ensuring the inclusion of their respective types.

Hours to be allocated: 60

Contact hours:

Independent study/self-guided study = 48 hours

Face-to-face learning = 32 hours

Total = 80

Reading list: The reading list for this module can be accessed at readinglists.uwe.ac.uk via the following link https://rl.talis.com/3/uwe/lists/1E04FAF9-1493-FE10-A739-4E142FB30146.html?lang=en-gb&login=1

Part 4: Assessment

Assessment strategy: This module assessment is split into two assessment tasks (Exam, Labs work):

A final 2-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.

The practical assignment should be completed individually (i.e. this is not group work).

A series of 3 practical tasks (labs), exploring basic principles of functional programming using HASKELL programming language. An application and its source code should be provided by students.

Resits: will be like for like.

Assessment tasks:

Examination (First Sit)

Description: Examination (2 hours)

Weighting: 50 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO3

Laboratory Report (First Sit)

Description: A series of 3 practical tasks (labs), exploring basic principles of functional programming using HASKELL programming language An application and its source code should be provided to the teaching staff in form of report.

Weighting: 50 %

Final assessment: No

Group work: No

Learning outcomes tested: MO2, MO3

Examination (Resit)

Description: Examination (2 hours)

Weighting: 50 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO3

Laboratory Report (Resit)

Description: A series of practical tasks (labs), exploring basic principles of functional programming using HASKELL programming language. An application and its source code should be provided to the teaching staff in form of report.

Weighting: 50 %

Final assessment: No

Group work: No

Learning outcomes tested: MO2, MO3

Part 5: Contributes towards

This module contributes towards the following programmes of study:

Computer Science and Software Development {Double Degree} {Foundation} [TSI] BSc (Hons) 2022-23

Computer Science and Software Development {Double Degree} [Feb][PT][TSI][5yrs] BSc (Hons) 2021-22

Computer Science and Software Development {Double Degree} [Oct][PT][TSI][5yrs] BSc (Hons) 2021-22