

Course Unit	Programming Technologies		Field of study	Computer Science	
Master in	Informatics		School	School of Technology and Management	
Academic Year	2023/2024	Year of study	1	Level	2-1
Type	Semestral	Semester	2	ECTS credits	6.0
Code	5060-710-1205-00-23				
Workload (hours)	162	Contact hours	T -	TP 60	PL -
			TC -	S -	E -
			OT -	O -	

T - Lectures; TP - Lectures and problem-solving; PL - Problem-solving, project or laboratory; TC - Fieldwork; S - Seminar; E - Placement; OT - Tutorial; O - Other

Name(s) of lecturer(s) Maria João Tinoco Varanda Pereira

Learning outcomes and competences

At the end of the course unit the learner is expected to be able to:

1. have knowledge on concepts, programming techniques and models of computer programming;
2. have an overview of the various programming paradigms with emphasis on its suitability to each class of problems;
3. know basic concepts about logic and functional programming;
4. have notions of automatic construction of language processors for creating solutions based on domain specific languages.

Prerequisites

Before the course unit the learner is expected to be able to:
Have knowledge on imperative paradigm concepts.

Course contents

In this curricular unit, the main characteristics of the various types of programming languages are analyzed and a classification system is presented within the scope of the paradigm they implement. In addition to talking about the logical and functional paradigm as a complement to the training obtained in the Degree, concepts of implementation of specific domain languages and automatic construction of language-based systems are also addressed.

Course contents (extended version)

1. Study of Programming Paradigms
 - History of programming languages;
 - Basic concepts of programming;
 - Language Classification System;
 - Language Assessment: readability and processing criteria;
 - More general paradigms: Imperative Paradigm, Logical Paradigm, Functional Paradigm;
 - More specific paradigms: Object-oriented, Event-oriented and Aspect-oriented
2. Language Processors
 - Lexical, syntactic and semantic analysis;
 - Automata, Grammars, and Attributes;
 - Automatic construction of processors using a CC tool.
3. Logical Paradigm
 - Notions of logical predicates and inference machine (search and backtracking);
 - Practical examples in Prolog;
4. Functional paradigm
 - Values and types; Functions; Pattern Recognition;
 - Type classes and overloading; input / output operations; standard Haskell classes;
 - Practical exercises in Haskell.

Recommended reading

1. Programming Languages: Principles and Paradigms, Maurizio Gabbriellini, Simone Martini, Springer, 2010.
2. Prolog Programming; Success in a Day, Sam Key, Kindle Edition, 2015.
3. Programming in Haskell, Graham Hutton, Kindle Edition, 2016.
4. PLY (Python Lex-Yacc), David Beazley, <https://www.dabeaz.com/ply/ply.html>, accessed at Fev 2024
5. An effective way to teaching Language Processing Courses, Maria João Varanda Pereira, Nuno Oliveira, Pedro Henriques, pp. 131-152, IGI Global, 2014.

Teaching and learning methods

This course is divided into theoretical lectures and practical lectures. In the theoretical lectures programming paradigms subject is exposed using some practical examples and asking for student participation. In practical lectures the student is invited to use specific tools to solve the exercises in a computing environment.

Assessment methods

1. Alternative 1 - (Regular, Student Worker) (Final, Supplementary)
 - Practical Work - 100% (4 practical assignments)
2. Alternative 2 - (Regular, Student Worker) (Special)
 - Final Written Exam - 100%

Language of instruction

1. Portuguese
2. English

Electronic validation

Maria João Tinoco Varanda Pereira	Tiago Miguel Ferreira Guimaraes Pedrosa	José Eduardo Moreira Fernandes	José Carlos Rufino Amaro
22-02-2024	14-03-2024	15-03-2024	16-03-2024