Curricular Unit Form (FUC)

Course:	**FIRST CYCLE IN MECHANICAL ENGINEERING**					
Curricular Unit (UC)	**Introduction to Programming**				Mandatory	**X**
					Optional	
Scientific Area:	**Energy and Control Systems**					
Year: **1st**	Semester: **1st**	ECTS:**4,0**		Total Hours: **3,0**		
Contact Hours:	T:	TP:**45,0**	PL:	S:	OT:	TT: **45,0**
Professor in charge		Academic Degree /Title		Position		
Fernando Paulo N. F. Carreira		**MSc**		**Assistant Professor**		

T- Theoretical ; TP – Theory and practice ; PL – Laboratory ; S – Seminar ; OT –Tutorial ; TT – Total of contact hours

Entry into Force	Semester: **Winter**	Academic Year: **2010/2011**

Objectives of the curricular unit and competences (max. 1000 characters)

- Meet the objectives of programming and its use in the context of Mechanical Engineering.
- Understand different types of variables and learn to manipulate them.
- Know how to use elementary functions and selection and repetition structures.
- Learn to develop algorithms in a structured form.
- Have contact with recent software tools for programming and development of algorithm in Mechanical Engineering.

Syllabus (max. 1000 characters)

1. Introduction to computing. Historical introduction to computing. Processing units and communication infrastructure. Operating systems. High-level and low-level languages.
2. Algorithmic. Theoretical concepts on algorithms: Algorithm, Pseudo-language, flowcharts. Data types and variables. Arithmetic and logical expressions; Sequential structures; repetition structures (*repeat-until*, *while* and *for*) and selection structures (*if-then -else* and *switch-case*). Implementation of Algorithms.
3. Programming in MatLab. The environment of platform; homogeneous and heterogeneous variables. Vectors and matrices; Manipulation of indexed variables. String manipulation. Editing programs. Routines and functions; Preparation of graphics; Programming using objects; Construction of graphical environments; Programming with blocks; Development of programs in Matlab applied to engineering problems.

Demonstration of the syllabus coherence with curricular unit's objectives (max. 1000 characters)
• The first chapter allows students to know the need to have developed programming languages as platforms for automated calculation. It is also focused on the connection of the hardware (I / O devices, processor and memory) with software (inputs, outputs, logical and arithmetic processing and data storage). • In the second chapter are taught the fundamental concepts of algorithms, making known the various data types and the main programming structures. This chapter challenges students to structure their thinking, developing algorithms that solve computational problems. • The applicability of the algorithms is made in the third chapter, where algorithms of problems applied to engineering are implemented in a programming language (M-code). For a correct application of the algorithm in the programming language, students need to write the instructions respecting the syntax M and sequence the instructions in a logical and coherent.

Teaching methodologies (including evaluation) (max. 1000 characters)
The unit is divided in theoretical and practical classes, were 40% of classes taught in the classroom and 60% in the lab. The assessment is done through a project (NP), which is considered pedagogically fundamental work, and continuous evaluation assessments (NAC) or a final exam (NE) for the students who cannot perform continuous evaluation. The final classification is calculated by the following equation: NF = 0.4 * (NAC or NE) + 0.6 * NP Continuous assessments consists of two practical work with the following distribution: • Practical assessment no. 1 - 25% of NAC. • Practical assessment no. 2 - 75% of NAC. Essential conditions for approval: • Submit all deliverables within the time limits. The components delivered outside the stipulated date (until the day of the examination at the time of appeal) will have a penalty of 25% in its classification. • Approval by continuous evaluation requires: ○ The minimum frequency classes is 75% (to date before each assessment) ○ Classification in all the evaluations ≥ 8 ○ Weighted average of the continuous evaluations ≥ 9.5 • Must have: ○ Classification in the project ≥ 9.5 ○ Classification in the continuous assessment and final exam ≥ 9.5 ○ The final classification, rounded to the units, should be positive (NF ≥ 10)

Demonstration of the teaching methodologies coherence with the curricular unit's objectives (max. 3000 characters)
• Theory and practical classes: 40% in classroom and 60% in IT Lab. • During the first week, the lectures alternate with practical sessions, conducted in the IT Lab, allowing students to practice the manipulation of variables and algorithmic structures. • After introducing the basic concepts, exercises are performed where students have to structure their thinking in order to develop algorithms and implement them using the syntax of the programming language. • The practical assessments evaluate the theoretical knowledge acquired in algorithms and their application in a specific syntax such as: concepts of variable and algorithm, data types, selection and repetition structures, as well as the ability to apply them in developing small applications. • The project allows students to work together developing a computational algorithm with application in the field of Mechanical Engineering. From the problem analysis, data structures and algorithm development to implement in a programming language. • The final exam is an alternative way to continuous evaluation; either for lack of notice required for approval, either by lack of attendance at classes, a factor that impedes the realization of continuous assessment.

Main Bibliography (max. 1000 characters)
• Brassard, G.; Bratley, P. (1996) *Fundamentals of Algorithms*. Prentice-Hall. ISBN-13 9780133350685 • Thomas, H. C.; Ronald L. R.; Charles E. L.; Clifford S. (1999) *Introduction to Algoritms*. MIT Press. 1999. ISBN-13 978-0262033848 • Morais, V. e Vieira, C. (2006) *Matlab 7&6 - Curso Completo*. FCA. ISBN 9789727223541 • Chapman, S. J. (2003) *Programação em Matlab para engenheiros*. Thomson. ISBN 8522103259 • Anjo, B.; Fernandes, A.J.; Carvalho, R.; Simões, A. (2003) *Curso de Matlab*. Principia. ISBN 9728818084. • Carreira, F. e Silva, P. (2007) *Introdução ao Cálculo e Programação em Matlab*. ISEL. Disponível na plataforma Moodle da Unidade Curricular: http://moodle.isel.pt/dem