

Computer Science

Code: 104524
ECTS Credits: 6

Degree	Type	Year	Semester
2503743 Management of Smart and Sustainable Cities	FB	1	1

The proposed teaching and assessment methodology that appear in the guide may be subject to changes as a result of the restrictions to face-to-face class attendance imposed by the health authorities.

Contact

Name: Carles Ferrer Ramis
Email: Carles.Ferrer@uab.cat

Use of Languages

Principal working language: catalan (cat)
Some groups entirely in English: No
Some groups entirely in Catalan: Yes
Some groups entirely in Spanish: No

Teachers

Raul Aragonés Ortiz

Prerequisites

There are not.

Objectives and Contextualisation

In this area, the basic concepts related to Information Technology and Communications (ICT) as tools for the development of city management applications will be introduced, as well as basic notions of algorithmics and application programming.

Competences

- Critically analyse work carried out and demonstrate a desire to improve.
- Generate innovative and competitive proposals in professional activity.
- Solve urban management problems using knowledge, methodology and procedures for the design and implementation of computer applications for different types of environment (web, mobile, cloud) and different paradigms.
- Students must be capable of applying their knowledge to their work or vocation in a professional way and they should have building arguments and problem resolution skills within their area of study.
- Students must be capable of communicating information, ideas, problems and solutions to both specialised and non-specialised audiences.
- Students must develop the necessary learning skills to undertake further training with a high degree of autonomy.

Learning Outcomes

1. Critically analyse work carried out and demonstrate a desire to improve.

2. Generate innovative and competitive proposals in professional activity.
3. Students must be capable of applying their knowledge to their work or vocation in a professional way and they should have building arguments and problem resolution skills within their area of study.
4. Students must be capable of communicating information, ideas, problems and solutions to both specialised and non-specialised audiences.
5. Students must develop the necessary learning skills to undertake further training with a high degree of autonomy.
6. Understand the basic architecture of computer applications in distinct types of environments (web, mobile, cloud).
7. Understand the basic principles for the development of computer applications in the management of cities.
8. Use basic programming structures to solve simple problems related to the management of cities.

Content

1. Computer applications in different types of environments (web, mobile, cloud) for the management of cities: concepts and examples.
2. Basic concepts of computers: structure, programming languages, operating systems, communications, interconnection of systems.
3. Algorithms and programming: concept and representation of an algorithm. modular design.
4. Basic data types.
5. Basic programming structures.
6. Representation of data.
7. Data input and output.

Methodology

The teaching methodology will be based on three types of activities:

- Guided activity: theoretical classes, laboratory, and exercise analysis.
- Supervised activity: attendance to tutorials and completion of exercises with scheduled follow-up.
- Autonomous activity: part of student study and case resolution, individually or in groups.

In order to be able to perform a correct assessment of the transversal competencies corresponding to the subject, the students will be proposed to carry out a joint work. This activity will allow them to develop the transversal competencies related to group work (T01), becoming responsible for the tasks assigned, respecting the role of the different members of the team, and evaluating critically the work carried out (T05) among them.

The preferred form of communication with students will be the virtual campus combined with the institutional mail of the UAB.

Annotation: Within the schedule set by the centre or degree programme, 15 minutes of one class will be reserved for students to evaluate their lecturers and their courses or modules through questionnaires.

Activities

Title	Hours	ECTS	Learning Outcomes
Type: Directed			

Autonomous work	76	3.04	1, 2, 8
Exercises and laboratories	24	0.96	6, 7, 5, 4
Theory classes	26	1.04	6, 7, 3, 8

Assessment

1. Evidence of continuous evaluation

There are two tests that include the seven blocks of matter (1,2,3 in the first test and 4,5,6,7 in the second test).

Continuous evaluation tests	Weight note continuous assessment	Minimum mark to make an average
1-2 Computer applications and basic systems	50%	3,5
3-7 Algorithms, data, structures, and representation	50%	3,5

2. Final evaluation mark

Final mark	Weight final mark
Continuous evaluation	60%
Class Picks	10%
Laboratory	30%

3. It is considered approved anyone:

- have a final mark equal to or greater than 5 and
- have approved the laboratory activities (minimum 5) i
- there is no evidence of the continuous evaluation below the minimum mark (4.0) to do the average.

4. Assessment of practices

There will be a total of 5 laboratory activities where the algorithm will be asked to bring home prepared for each one of them, which will count 10% of the laboratory activities mark.

5. Class picks cannot be retrieved.

6. There will be a final exam of the two blocks of theory aimed at recovering the not surpassed part of the continuous evaluation.

7. At the beginning of the academic year, if possible, it will be notified if there is a validation of laboratory activities. In the case of being, the validation of laboratory activities only will realize to the students who request it and have approved the laboratory activities in the previous course. The weight of the continuous evaluation in the final mark, in the students with the validation of laboratory activities, becomes 90%.

8. Continuous evaluation dates are set at the beginning of the course and do not have alternative recovery date in case of non-attendance. If there is any change in programming due to the adaptation to possible incidents, the virtual campus will always be informed about these changes.

9. Notwithstanding other disciplinary measures deemed appropriate, and in accordance with the current academic regulations, irregularities committed by a student that can lead to a variation of the qualification will be classified by zero (0). For example, plagiarizing, copying, copying, ..., an evaluation activity, will imply suspending this evaluation activity with zero (0). Assessment activities qualified in this way and by this procedure will not be recoverable. If it is necessary to pass any of these assessment activities to pass the subject, this subject will be suspended directly, without an opportunity to recover it in the same course.

10. Non-evaluable cases

In case no delivery is made, it will not be included in any laboratory session and no exam will be carried out, the corresponding grade will be "not evaluable". In any other case, "not presented" counts as a 0 for calculating the weighted average, which will be a maximum of 4.5. Then, participation in an activity evaluated implies that "not presented" in other activities such as zeros are taken into account. For example, an absence in a laboratory session involves a note for that activity.

11. Pass de course with honors

To pass the course with honors will be awarded to those who obtain a mark greater than or equal to 9.5 in each part, up to 5% of those enrolled in descending order of final grade. At the discretion of the teaching staff, they may also be granted in other cases.

Assessment Activities

Title	Weighting	Hours	ECTS	Learning Outcomes
Evaluation tests	60%	4	0.16	6, 7, 2, 5, 4, 3, 8
Laboratory activities	30%	10	0.4	6, 7, 5
Supervised activities	10%	10	0.4	1, 2, 4

Bibliography

- A. Prieto, A. B. Prieto. "Conceptos de informática". Editor Mc Graw Hill, 2005.
- Mark Lutz. Learning Python, Fourth Edition. Editor O'Reilly Media, Inc., 2009.
- Guía de uso del MIT App Inventor. Escuela superior de informática de Castilla la Mancha.
<http://webpub.esi.uclm.es/img/upload/plugin/ESI-TechLab-AppInventor2-2015beta.pdf>

Software

Python IDLE

MIT App Inventor