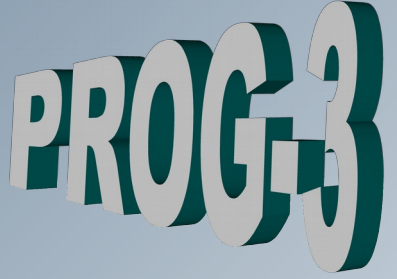


PROGRAMMING 3

Academic Year 2016-2017

Degree in Computer Engineering (34012)

Course Description

Version 20160903

lsi Departamento de Lenguajes y Sistemas Informáticos

Universitat d'Alacant
Universidad de Alicante

Instructors

All belong to Departament de Llenguatges i Sistemes Informàtics (DLSI)

- **Pedro J. Ponce de León** (Theory and practicals) *Coordinator*
- **José Gonzalo Alcalá Aparicio** (Practicals)
- **Francisco Moreno Seco** (Practicals)
- **David Rizo Valero** (Theory and practicals in Valencian)
- **Juan Antonio Pérez Ortiz** (Theory and practicals, **ARA group**)

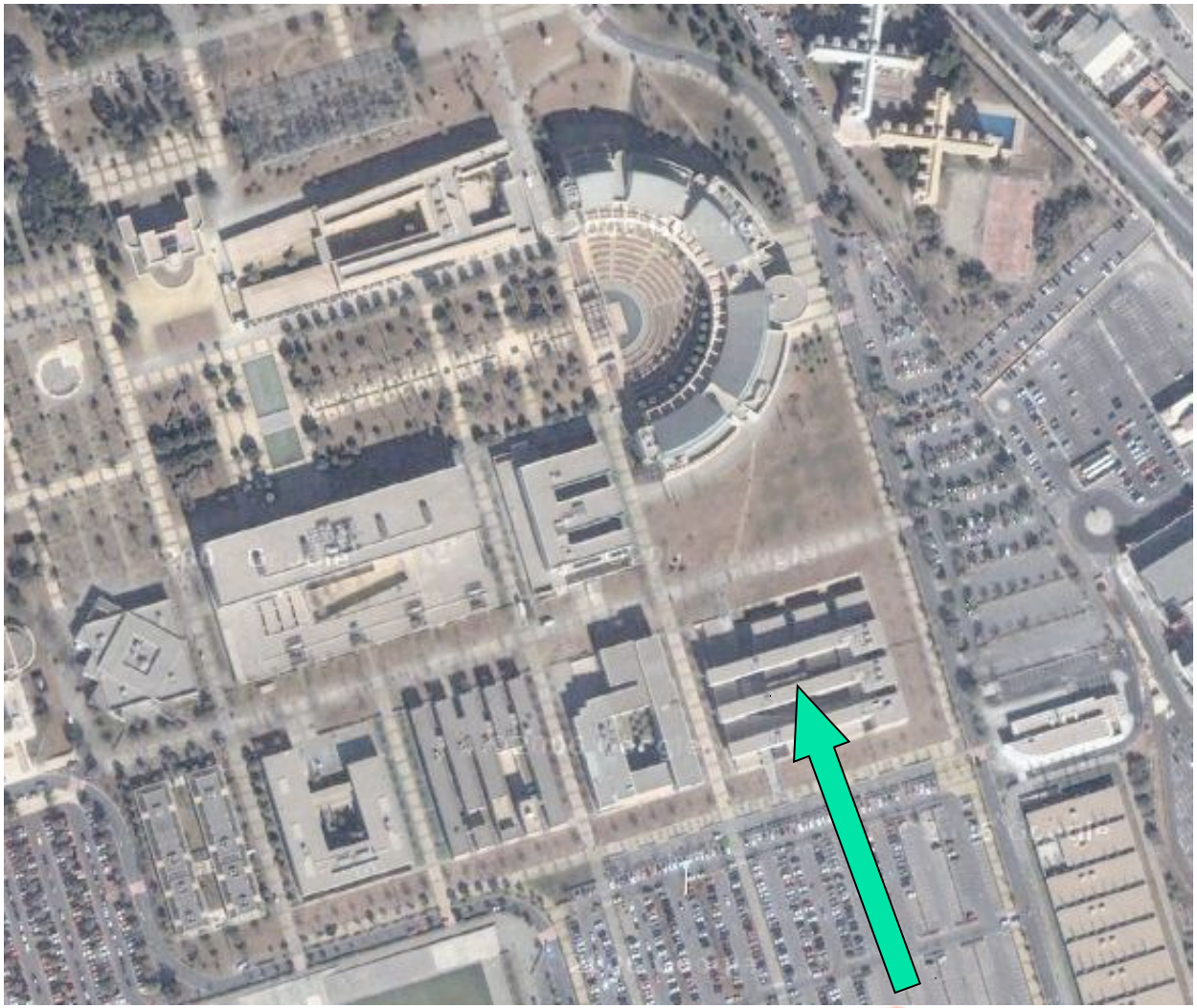
- Important note: electronic communication solely through campus virtual for all teachers; office hours by previous arrangement

Offices, office hours, arrangements, etc: www.dlsi.ua.es

ARA group

- Official version of all the teaching materials for the students enrolled in the ARA group is the English one when available. You will be graded according to the information therein contained, and not that in the Spanish version.
- Correct time management is important for both students and professors! Consequently, you are required to request an appointment before visiting the professor in his office hours. Use the form available here for that: http://www.dlsi.ua.es/alumnes/?id=eng

Department of Computer Languages and Systems (DLSI)



Street view

Syllabus

Campus Virtual → Recursos de aprendizaje → Guía docente

Class hours, objectives, schedule, grading, textbooks and other course materials, links

"PROGRAMMING 3 [PROG-3] provides a thorough coverage of a fundamental topic in the technical knowledge and practical skills of a computer engineer working in software engineering:

object oriented programming (OOP),

the dominant programming paradigm in almost every computer science area".

(TIOBE index, Transparent Language Popularity Index, LangPop)

Context

Programming 1/2 (1st year)

---- PROGRAMMING 3 (2nd year) ----

- **Programming and Data Structures,**
  Algorithm Analysis and Design (2nd year, 2nd semester)
  Abstract data types: definition, formalization, usage...; algorithm analysis, algorithmic paradigms, searching, sorting...

  Advanced Tools for Application Development (2nd year, 2nd semester)
  User interfaces, object code reuse, licenses...

- **Programming Languages and Paradigms (2nd year, 2nd semester)**
  Programming paradigms, functional programming, logic programming...

- **Analysis and Specification of Software Systems,**
  Software System Design,
  Planning and Testing Software (3rd year)
  Requirements analysis, modeling languages, design, software patterns, cost estimation, configuration management...

Topics

- Topic 1: **Introduction to the OO paradigm**

Subtopic 1. Introduction to the OO paradigm. Classes and objects.
Subtopic 2. Relationships between objects.

- Topic 2. **Fundamental concepts**

Subtopic 3. Error management
Subtopic 4. Implementation inheritance
Subtopic 5. Compilation, linking time and memory management in OO languages.

Topics

- Topic 3. **Fundamental concepts 2**

Subtopic 6. Polymorphism 1. Overloading and overriding.
Subtopic 7. Interface inheritance
Subtopic 8. Polymorphism 2. Genericity and reflection.

- Topic 4. **Software reuse and application maintenance**

Subtopic 9. Frameworks and libraries
Subtopic 10. Code maintenance

- Topic 5. **Principles of OO design**

Subtopic 11. Principles of OO design

Theory Schedule

Week		Wednesday
1	12/9	Topic 1
2	19/9	Topic 1
3	26/9	Topic 2
4	3/10	Topic 3
5	10/10	Holidays
6	17/10	Topic 4
7	24/10	Topic 5
8	31/10	Topic 6
9	7/11	Topic 7 (1/2)
10	14/11	Topic 7 (2/2)
11	21/11	Topic 8
12	28/11	Topic 9
13	5/12	Topic 10
14	12/12	Topic 11
15	19/12	(Course review)

Practical Schedule

Weeks	Tentative Publishing Date	Number	Lab Assignment	Tentative Deadline
1-2	Sept. 12	P1	From C++ to Java	Friday Sept. 23
3-5	Sept. 26	P2	Relationships	Thursday Oct. 14
6-10	Oct. 17	P3	Simple inheritance + exceptions	Friday Nov. 11
10-13	Nov. 14	P4	Interface inheritance/ polymorphism	Friday Dec. 9
14-15	Dec. 12	P5	Reflection/ refactoring/ frameworks	Friday Dec. 23

Grading

- **Continuous assessment**
  - Programming assignments (50%)
    - Lowest grade allowed: 4
    - Attendance to lab sessions is mandatory (4 absences allowed)
- **Final exam** (50%)
    - Lowest grade allowed: 4

Minimum grades: grades in each section will be considered if they are equal or higher than the lowest grade allowed. Students with at least one grade below the boundary will not pass the course and their final grade will be the minimum between the calculated grade and 4.5.

Grading

- The procedure established for students to request a review of the exam mark will also be applicable in order to request a review for any other evaluation test.
- Second period of evaluation (July): both the programming assignments and the final exam may be retaken. If the student does not take one of them, the corresponding mark obtained in the first period of evaluation will prevail; conversely, taking any of the parts implies renouncing to the mark obtained in the first period of evaluation for that part.
- Assessing the programming assignments in the second period of evaluation: the student will have to submit the solutions of the assignments and take a programming examination on the same date as the final exam. Submission of the assignment, under the terms published by the professors on the Virtual Campus, will be mandatory, but the mark corresponding to the programming assignments will only be obtained from the programming examination.

Assignment Grading

5 lab assignments: object oriented programming incremental project

- **Weights for each assignment:**
  - P1 5%
  - P2 20%
  - P3 30%
  - P4 30%
  - P5 15%

  • Possible scenario: 10x0.05 + 10x0.2 + 8x0.3 + 0 (not submitted) + 3*0.15 = 5.35

- **Automatic evaluation**

  •Tests will be published after evaluation. Tests for an assignment will be used for the evaluation of the next one.

- Application domain: board game

More Information