Universidad de Alcalá

# TEACHING GUIDE

# Advanced programming

## Degree in
## Computer Engineering (GIC)
## Computer Science Engineering (GII)

## Universidad de Alcalá

---

## Academic Year 2021/2022

2nd Year - 2nd Semester (GIC+GII)

Approved by the School Board on July 6, 2021

# TEACHING GUIDE

| | |
|---|---|
| Course Name: | **Advanced programming** |
| Code: | **780014 (GIC+GII)** |
| Degree in: | **Computer Engineering (GIC)**<br>**Computer Science Engineering (GII)** |
| Department and area: | **Ciencias de la Computación**<br>**Computer Science** |
| Type: | **Compulsory (GIC+GII)** |
| ECTS Credits: | **6.0** |
| Year and semester: | **2nd Year - 2nd Semester (GIC+GII)** |
| Teachers: | José María Gutiérrez Martínez, Ana Castillo Martínez |
| Tutoring schedule: | Will be published at the beggining of the course |
| Language: | English |

# 1. COURSE SUMMARY

This course introduces new concepts and programming techniques that allow students to handle the creation of advanced applications that use concurrency, distribution and Mobile.
To reach this goals, the course will show the origins, evolution, present time and future of the tools, languages and theories behind concurrency.
Entry requirements: Students are recommended to have previously taken the Programming courses and data structures course.

# 2. SKILLS

### Basic, Generic and Cross Curricular Skills.

This course contributes to acquire the following basic, generic and cross curricular skills:

**en_CG4** - Ability to define, evaluate and select hardware and software platforms for the development and execution of computer systems, services and applications, in accordance with the knowledge acquired as set out in section 5, annex 2, of resolution BOE-A-2009 -12977.

**en_CG6** - Ability to conceive and develop centralized or distributed computer systems or architectures integrating hardware, software and networks in accordance with the knowledge acquired as set out in section 5, annex 2, of resolution BOEA-2009-12977.

**en_CB1** - That students have demonstrated to possess and understand knowledge in an area of study that is based on general secondary education, and is usually found at a level that, although supported by advanced textbooks, also includes some aspects that involve knowledge from the forefront of their field of study.

**en_CB2** - That the students know how to apply their knowledge to their work or vocation in a professional manner and possess the competencies that are usually demonstrated through the elaboration and defense of arguments and the resolution of problems within their area of study.

**en_CB3** - That students have the ability to gather and interpret relevant data (usually within their area of study) to make judgments that include a reflection on relevant social, scientific or ethical issues.

**en_CB4** - That students can transmit information, ideas, problems and solutions to both a specialized and non-specialized public.

**en_CB5** - That the students have developed those learning skills necessary to undertake further studies with a high degree of autonomy.

**en_TRU1** - Capacity of analysis and synthesis.

**en_TRU2** - Oral and written competencies.

**en_TRU3** - Ability to manage information.

**en_TRU4** - Autonomous learning skills.

**en_TRU5** - Team work.

### Specific Skills

This course contributes to acquire the following specific skills:

**en_CI8** - Ability to analyze, design, build and maintain applications in a robust, safe and efficient way, choosing the most appropriate programming paradigm and languages.

**en_CI11** - Knowledge and application of the characteristics, functionalities and structure of Distributed Systems, Computer Networks and Internet and to design and implement applications based on them.

**en_CI14** - Knowledge and application of the fundamental principles and basic techniques of parallel, concurrent, distributed and real-time programming.

### Learning Outcomes

After succeeding in this subject the students will be able to:

**RA1**. Evaluate the effect of programming languages capabilities and limitations on computer systems creation process.

**RA2**. Describe the evolution of programming languages, paradigms available and their main characteristics.

**RA3**. Analyze the effect on information systems design of the evolution of programming languages, paradigms available and their main characteristics.

**RA4**. Apply parallel, concurrent and distributed programming, with the knowledge of their advantages, disadvantages and main algorithm, to the design of more efficient systems.

## 3. CONTENTS

Topic 1: Programming paradigms. Programming languages history. Short description of some paradigms. Comparison between paradigms, utility and influence over programming languages.

Topic 2: Introduction to concurrency. Historic evolution and terminology. Concurrency problems. Architectures that allows concurrency. Shared variables. Distributed memory. Justice hypothesis. Safety and vitality properties.

Topic 3: Shared memory concurrency: Monitors. Cooperate and compete processes. Examples. Algorithms, Active waiting, Loks and conditions.

Topic 4: Shared memory concurrency in Java: Semaphores, Critical Regions, Conditional CR and monitors

Topic 5: Java concurrency. JSR166 specification. Concurrency and utilities.

Topic 6: Architectures and patterns for concurrent programming. Tasks and thread pools.

Topic 7: Introduction to distributed programming. Java sockets. Client/Server paradigm.

Topic 8: Distributed systems with Java objects. RMI (Remote Method Invocation). Synchronization and concurrency patterns.

| Contents Blocks | Total number of hours |
|---|---|
| **Module 1. Programming paradigms and introduction to concurrent programming (Topics 1, 2 and 3)** | 2 ECTS |
| **Module 2. Advanced concurrency and utilities, tasks and thread pools (Topics 4, 5 and 6)** | 2.5 ECTS |
| **Module 3. Distributed concurrency (Topics 7 and 8)** | 1.5 ECTS |

# 4. TEACHING - LEARNING METHODOLOGIES. FORMATIVE ACTIVITIES.

## 4.1. Credits Distribution

| | |
|---|---|
| Number of on-site hours: | 58 hours (56 hours on-site +2 exams hours) |
| Number of hours of student work: | 92 |
| Total hours | 150 |

## 4.2. Methodological strategies, teaching materials and resources

The course contents previously described shall be taught in the following ways:

- Taught theory classes
- Supervised practical classes: problem solving in class.
- Supervised practical labs.
- Tutorials: individual or group.

In addition, depending on the nature of the work, the students may make use of the following study methods, as well as others:

- Individual realization of coursework but with information input and management as part of a team.
- Exchange of information, problems and doubts which arise during individual work with course mates.
- Organization and production of published journal articles alongside oral presentations and discussions on the results.
- Use of the Virtual Learning Platform as a principal form of access to all activities and subject materials.

Class contact hours:

- In class: Presentation and discussion of core subject knowledge. Planning and theoretical solving and problems and related hypotheses. Oriented towards the teaching of subject specific skills, especially those related to the key concepts and practices of the imperative programming paradigm.
- In practical labs: Planning and development of practical exercises which allow problems to be

solved and hypotheses to be analyzed, contributing to the development of analytical and critical reasoning skills as well as an understanding of problem solving methods. These will serve as a basis for acquiring the general skills described in part 2 of this guide.

Outside of class:

- Analyzing and learning course contents, solving of problems, consulting the bibliography, individually preparing coursework, sitting exams and self-evaluation. Oriented especially towards developing personal organization skills and planning work individually or as part of a team.
- Tutorials: Individuals and group guidance throughout the learning process. Students may attend in person or online.

Materials and resources:

- Reference bibliography of core and further reading on the subject.
- Personal computers.
- Development environments and accompanying user guides
- Internet connection.
- Virtual Learning Platform and accompanying user guides.
- Projectors.

# 5. ASSESSMENT: procedures, evaluation and grading criteria

Preferably, students will be offered a continuous assessment model that has characteristics of formative assessment in a way that serves as feedback in the teaching-learning process.

## 5.1. PROCEDURES

The evaluation must be inspired by the criteria of continuous evaluation (Learning Assesment Guidelines, LAG, art 3). However, in compliance with the regulations of the University of Alcalá, an alternative process of final evaluation is made available to the student in accordance with the Learning Assesment Guidelines as indicated in Article 10, students will have a period of fifteen days from the start of the course to request in writing to the Director of the Polytechnic School their intention to take the non-continuous evaluation model adducing the reasons that they deem convenient. The evaluation of the learning process of all students who do not apply for it or are denied it will be done, by default, according to the continuous assessment model. The student has two calls to pass the subject, one ordinary and one extraordinary.

**Ordinary Call**

Continous Assessment:

The continuous assessment system consists of: a laboratory exercise, a programming exercise in the laboratory and written theoretical/ practical tests which assess taught skills. These exercise will be submitted via the Virtual Learning Platform.

Grading system for Continuous Assessment.

- Each theoretical CAT aims to assess knowledge of the material covered in each test. Students will acquire this knowledge by attending classes, studying the course materials prepared by teachers, doing the further reading suggested for each topic, searching for additional material and doing the exercises suggested by teachers or acquired by students.
- Practical CAT involves the creation of a completed application which must apply all the knowledge and skills acquired during the course. Also, there will be an examination or oral defense about the content of the CAT.

- The dates of the CAT examinations shall be given to students in the first days of the course in the 'Course Schedule' which can be found on the Virtual Learning Platform. All deadlines and events taking place during the course can be found there.

Weighting of continuous assessments (PEI) in the overall grade:

The final grade is comprised of 60% theory and 40% practice. The final weighting of each exam/ practical is as follows:

| **Practical CATs** | **% of overall grade** |
|---|---|
| PL | 40 |

| **Theoretical CATs** | **% of overall grade** |
|---|---|
| PEI1 | 10 |
| PEI2 | 35 |
| PEI3 | 15 |

Laboratory practice (global result) must be passed to pass the course. Failing the lab means that the student did not get all the expected competences. Appropriate mechanisms will be estabished to ensure the students can pass the labs in case of final examination and extraordinary call.

Assessment through final exam:

The end of term examination is only available in certain specified cases and must be requested by students who meet the criteria and who have been granted permission by school administration in accordance with the applicable regulation of the University of Alcalá. Students who take the summative end of term examination will sit an exam which will cover all the theoretical aspects of the course. They must also provide all the practical laboratory work that has been handed in CATL. If the students do not pass the lab assignment, they will receive the mark of the lab as final mark of the course.

### Extraordinary Call

Examination re-sits shall be held during the month of June for students who failed to pass both continuous assessments and the final exam. It consists of an examination in the same format as the final exam, with the same type of submissions (with a different focus).

## 5.2. EVALUATION

### EVALUATION CRITERIA

The assessment criteria measure the level in which the competences have been acquired by the student. For that purpose, the following are defined:

**CE1.** The student knows the history and the characteristics of programming languages and he is able to describe differences between programming paradigms.

**CE2.** The student understand concurrent programming, its types and evolution from sequential programming.

**CE3.** The student knows the principles of concurrent programs design.

**CE4.** The student have acquired basic knowledge about concurrent programs coordination.

**CE5.** The student have acquired the advanced single memory concurrent application design knowledge.

**CE6.** The student understand the need of distributed concurrent programming and the differences with single memory concurrency.

**CE7.** The student knows to program in a language supporting distributed concurrency using sockets or RPC.

**CE8.** The student knows to develop a single memory concurrent system using any available mechanism of programming languages.

**CE9.**The student have acquired basic knowledge about advanced concurrent programming.

## GRADING CRITERIA

In the ordinary call-continuous assessment the relationship between the competences, learning outcomes, criteria and evaluation instruments is as follows.

| Skills | Learning Outcomes | Evaluation criteria | Grading Tool | Contribution to the final mark |
|---|---|---|---|---|
| CG4,CI8 | RA1, RA2, RA3 | CE1, CE2 | PEI1 | 100% |
| CG4,CG6,CI14 | RA1, RA3, RA4 | CE3, CE4, CE5, CE8, CE9 | PEI2 | 35% |
| CG4, CG6, CI11, CI14 | RA4 | CE6, CE7 | PEI3 | 15% |
| CG4, CG6, CI11, CI14 | RA1, RA3, RA4 | CE3, CE5, CE6, CE7, CE8, CE9 | PL | 40% |

In the ordinary call-final evaluation, the relationship between the competences, learning outcomes, criteria and evaluation instruments is as follows.

| Skills | Learning Outcomes | Evaluation criteria | Grading Tool | Contribution to the final mark |
|---|---|---|---|---|
| CG4, CG6, CI8, CI11, CI14 | RA1, RA2, RA3, RA4 | CE1, CE2, CE3, CE4, CE5, CE6, CE7, CE8, CE9 | PEF | 60% |
| CG4, CG6, CI8, CI11, CI14 | RA1, RA3, RA4 | CE3, CE5, CE6, CE7, CE8, CE9 | PL | 40% |

Extraordinary call

In the case of the extraordinary call, the same percentages that have been established in the case of the evaluation by means of a final exam will be maintained, giving the option of making the PL or maintaining the mark obtained in the EL (continuous evaluation) or in the PEF (final evaluation), according to the student's decision. In any case, the PL will be made by those students who have not done it in the final exam option in the ordinary call.

# 6. BIBLIOGRAPHY

## 6.1. Basic Bibliography


- "Java Concurrency in practice". Goetz, Brian / Peierls, Tim / Bloch, Joshua / Bowbeer, Josheph / Holmes, David / Lea, Doug. 2007. Adison Wesley
- "Java in Distributed Systems". Boger, Marko. 2001. Wiley


## 6.2. Additional Bibliography


- Concurrent programming
  - "Concurrent Programming in Java™: Design Principles and Pattern, 2nd Edition". Lea, Doug. 2000. Adison Wesley
  - “Principles of Concurrent and Distributed Programming, Second Edition”. M. Ben-Ari. Addison-Wesley. 2006.
  - “The Art of Concurrency”. Clay Breshears. O'Reilly Media, Inc. 2009.
  - “Concurrency: State Models and Java Programs”. Jeff Magee & Jeff Kramer. John Wiley & Sons. 2006.
- Distributed programming
  - "Concurrent Systems. Operating Systems, Database and Distributed Systems". Bacon, J. 1998. Addison Wesley
  - "Object-oriented Reuse, Concurrency and Distribution". Atkinson, Colin. 1991. Addison Wesley
  - "Parallel Program Design". Chandy, Mani / Misra, Jayedev. 1996. Addison Wesley
  - "The Java Programming Language". Arnold, Ken / Gosling, James / Holmes, David. 2005. Addison Wesley
  - "Concurrent and Real-Time Programming in Java". Wellings, Andrew. 2004. John Wiley & Sons
  - "Concurrent Programming Concepts". Brinch-Hansen, P. 1973. ACM Computing Survey
  - "Java Distributed Computing". Farley, J. 1998. O'Reilly & Associates, Sebastopol
- Programming
  - “Exercises in Programming Style”. Cristina Videira Lopes. CRC Press. 2014

## Disclosure Note

The University of Alcalá guarantees to its students that, if due to health requirements the competent authorities do not allow the total or partial attendance of the teaching activities, the teaching plans will achieve their objectives through a teaching-learning and evaluation methodology in online format, which will return to the face-to-face mode as soon as these impediments cease.