Universidad de Alcalá

TEACHING GUIDE

Programming Paradigms

Degree in
Information System Engineering

Universidad de Alcalá

Academic Year 2023/2024

3rd Year - 1st Semester

Approved by the School Board on June 12th, 2023

TEACHING GUIDE

Course Name:	**Programming Paradigms**
Code:	**581005**
Degree in:	**Information System Engineering**
Department and area:	**Ciencias de la Computación** **Computer Languages and Systems**
Type:	**Compulsory**
ECTS Credits:	**6.0**
Year and semester:	**3rd Year, 1st Semester**
Teachers:	Antonio García Cabot Eva García López
Tutoring schedule:	Consultar al comienzo de la asignatura
Language:	Spanish/English Friendly

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 these goals, the course will show the origins, evolution, present time and future of the tools, languages and theories behind concurrency.

To follow this course is mandatory to have competences in basic and object-oriented programming.

Java programming language is used throughout the course to show examples and to implement every type of program. Knowledge of this language is necessary to follow the course.

This course also introduces intelligent environments with the capability of communication and information processing that surrounds people in the present day. Those environments are ambient intelligence or pervasive computing, mobile computing and ubiquitous computing. All of them will be defined through the presentation of their characteristics and complete development projects will be offered.

The creation of invisible technology using sensors and actuators, based on simple boards will be studied. Methodologies for these kinds of developments will be presented and used among the projects. Business models will be analyzed and the best option for every project will be selected.

2. SKILLS

Basic, Generic and Cross Curricular Skills.

This course contributes to acquire the following generic 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_CI6 - Knowledge and application of the basic algorithmic procedures of computer technologies to design solutions to problems, analyzing the suitability and complexity of the proposed algorithms.

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.

en_CI15 - Knowledge and application of the fundamental principles and basic techniques of intelligent systems and their practical application.

en_CI16 - Knowledge and application of the principles, methodologies and life cycles of software engineering.

Learning Outcomes

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

RA1. Analyze how the capabilities and limitations of the most common programming languages affect the creation of computer systems.
RA2. Describe the evolution of programming languages, the different paradigms available today and their main characteristics.
RA3. Critically analyze the evolution of programming languages, the different paradigms available today and their main features in relation to their effect on the design decisions of information systems.
RA4. Know the distributed, concurrent, parallel and real-time programming, identifying their fundamental algorithms and the advantages and disadvantages of each paradigm, to design more effective or efficient systems from the point of view of performance, efficiency for the user or others.

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

Topic 6: Distributed systems with Java objects. RMI (Remote Method Invocation). Synchronization and concurrency patterns.

Topic 7: Developing projects for ubiquitous technology, smart environments, Internet of Things (IoT).

Contents Blocks	Total number of hours
Programming paradigms. Topic 1	0,5 ECTS
Concurrent programming. Topics 2, 3 and 4.	2,5 ECTS
Distributed concurrency. Topics 5 and 6	2 ECTS
Developing projects for ubiquitous technology, smart environments, Internet of Things (IoT). Topic 7	1 ECTS

4. TEACHING - LEARNING METHODOLOGIES. FORMATIVE ACTIVITIES.

4.1. Credits Distribution

Number of on-site hours:	60 hours (56 hours on-site +4 exams hours)
Number of hours of student work:	90
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 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

Continuous 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 exercises will be submitted via the Virtual Learning Platform.

The grading system for Continuous Assessment.

- Each theoretical CAT (T1, T2 and T3) 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 CAL (PL) involves the creation of a completed application that 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.

The weighting of continuous assessments (CAT) 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
T1	30
T2	20
T3	10

To pass the course, students must take exams PEC1T and PEC2T and pass the evaluation of the competences related to both exams. Students acquire such competencies satisfactorily if their weighted mark is equal or higher than 40% of the maximum possible mark.

Laboratory assignment (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 established to ensure the students can pass the labs in case of final examination and extraordinary call.

To pass the course, students must obtain a final weighted mark equal to or greater to 5 out of 10.

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 the 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 that will cover all the theoretical aspects of the course. They must also provide all the practical laboratory work that has been handed in CATL (PL).

To pass the course, students must take the theoretical exam and pass the evaluation of the competences related to that exam. Students acquire such competencies satisfactorily if their mark in the theoretical exam is equal or higher than 40% of the maximum possible mark.

If the students do not pass the lab assignment, they will receive the mark of the lab as the final mark of the course.

To pass the course, students must obtain a final weighted mark equal to or greater to 5 out of 10.

Extraordinary Call

The procedure will be the same as that described for the assessment by means of a final exam in the ordinary call.

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 actively participates in a development team in a non-conventional software development project.

GRADING CRITERIA

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

Skill	Learning Outcomes	Evaluation criteria	Grading Tool	Contribution to the final mark
CG4, CG6, CI6, CI14	RA1, RA2, RA3, RA4	CE1, CE2, CE3, CE4, CE7, CE8	CAT1	30%
CG4, CG6, CI11	RA3, RA4	CE5, CE6	CAT2	20%
CG4, CG6, CI15, CI16	RA4	CE9	CAT3	10%
CG4, CG6, CI6, CI11, CI14	RA1, RA3, RA4	CE4, CE5, CE6, CE7, CE8	CAL	40%

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

Skill	Learning Outcomes	Evaluation criteria	Grading Tool	Contribution to the final mark
CG4, CG6, CI6, CI11, CI14, CI15, CI16	RA1, RA2, RA3, RA4	CE1, CE2, CE3, CE4, CE5, CE6, CE7, CE8, CE9	PEF	60%
CG4, CG6, CI6, CI11, CI14	RA1, RA3, RA4	CE4, CE5, CE6, CE7, CE8	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.

6. BIBLIOGRAPHY

6.1. Basic Bibliography

- APUNTES DE LA ASIGNATURA, disponibles a través de la plataforma virtual.
- "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

- "Programación concurrente". Palma Martínez, José Tomás / Garrido Carrera, Mª del Carmen / Sánchez Figueroa, Fernando / Quesada Arencibia, Alexis. 2003. Thomson
- "Programación concurrente en Java". 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. Programación Distribuida
- "Sistemas Distribuidos. Conceptos y diseño". Coulouris, George / Dollimore, Jean / Kindberg, Tim. 2001. Adison Welsey
- "Aplicaciones Distribuidas en Java con RMI". Caballé, Santi / Xhafa, Fatos. 2008. Delta Publicaciones
- "Introduccion a la Programacion Paralela". Almeida, F. / Gimenez, D. / Mantas, Jose Miguel / Vidal, A.M. Paraninfo Cengage Learning. 2008.
- "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
- Ubiquitous Computing Fundamentals. 2010

- Designing Connected Products: UX for the Consumer Internet of Things. 2015
- Ubiquitous Computing and Intelligent Systems. 2016

Disclosure Note

During the evaluation tests, the guidelines set out in the Regulations establishing the Rules of Coexistence of the University of Alcalá must be followed, as well as the possible implications of the irregularities committed during said tests, including the consequences for committing academic fraud according to the Regulation of Disciplinary Regime of the Students of the University of Alcalá.