ANX-PR/CL/001-01
LEARNING GUIDE

SUBJECT

103000806 - Correctness By Construction

DEGREE PROGRAMME

10AR - Master Interuniversitario En Métodos Formales En Ingeniería Informática

ACADEMIC YEAR & SEMESTER

2021/22 - Semester 2

Index

Learning guide

1. Description

1.1. Subject details

Name of the subject	103000806 - Correctness By Construction
No of credits	6 ECTS
Type	Optional
Academic year ot the programme	First year
Semester of tuition	Semester 2
Tuition period	February-June
Tuition languages	English
Degree programme	10AR - Master Interuniversitario en Métodos Formales en Ingeniería Informática
Centre	10 - Escuela Tecnica Superior De Ingenieros Informaticos
Academic year	2021-22

2. Faculty

2.1. Faculty members with subject teaching role

Name and surname	Office/Room	Email	Tutoring hours *
Manuel Carro Liñares (Subject coordinator)	2303	manuel.carro@upm.es	F - 15:00 - 20:00 Please note that the office hours may change during the course. Please get in touch with the instructor to get an appointment.

Manuel De Hermenegildo Salinas	2212	manuel.hermenegildo@upm.es	Sin horario. Please get in touch with the instructor to get an appointment.

* The tutoring schedule is indicative and subject to possible changes. Please check tutoring times with the faculty member in charge.

3. Prior knowledge recommended to take the subject

3.1. Recommended (passed) subjects

The subject - recommended (passed), are not defined.

3.2. Other recommended learning outcomes

- Declarative programming

- First-order logic

- Programming experience (minimum 2 years)

- Formal proofs

- Reasoning about properties of algorithms

4. Skills and learning outcomes *

4.1. Skills to be learned

CE03 - Capacidad para utilizar técnicas y herramientas avanzadas, automáticas y asistidas, para verificar formalmente que un programa o sistema informático satisface las propiedades lógicas previamente especificadas.

CE07 - Capacidad para aplicar técnicas matemáticas a problemas informáticos relevantes, tales como el diseño de protocolos criptográficos seguros, la construcción de modelos formales del software, o el diseño de sistemas que aprenden automáticamente durante su ejecución.

CG07 - Capacidad para comprender trabajos de investigación y para crear nuevo conocimiento en el área de los métodos formales aplicados a la Ingeniería Informática.

4.2. Learning outcomes

RA5 - Effective use of rigorous software development techniques

RA3 - Knowledge of techniques for proving code correctness

* The Learning Guides should reflect the Skills and Learning Outcomes in the same way as indicated in the Degree Verification Memory. For this reason, they have not been translated into English and appear in Spanish.

5. Brief description of the subject and syllabus

5.1. Brief description of the subject

Software is becoming increasingly complex and responsible for critical tasks. Any technology aimed at ensuring the reliability and quality of software will be increasingly relevant, if not utterly necessary.

Only rigorous (e.g., mathematically sound) approaches can certify software with the highest possible assurance. These approaches include, among others, the use of specification languages, high-level programming languages (including equational, functional, and logic languages), the use of model checking and deductive verification, language-based approaches often interacting with theorem provers.

In this course we will give a hands-on introduction to rigorous software development methods that follow a *correctness-by-construction* approach. While the course is not heavy in theory, everyone is expected to have a good understanding of first-order logic and programming experience.

5.2. Syllabus

1. Introduction to Formal Methods: Proving Programs Correct

2. Fundamentals of Formal Methods: Specification, First-Order Logic, Proofs, Programs

3. Event-B Basics and the Rodin Tool

4. Sequential Systems

5. Event B: Mathematical Toolkit and Applications

6. Reactive Systems: Concurrency and Distribution

7. From Automated Deduction to Programming with Logic

8. Semantics and Advanced Features

9. CLP and Program Verification via Abstract Interpretation

6. Schedule

6.1. Subject schedule*

Week	Face-to-face classroom activities	Face-to-face laboratory activities	Distant / On-line	Assessment activities
1	**Introduction to formal methods and correctness by construction** Duration: 01:30 **Sample cases of formal development** Duration: 01:30			
2	**Event-B and related topics** Duration: 02:00 **Quizzes** Duration: 01:00			
3	**Event-B and related topics** Duration: 01:00			**Homework: solutions and discussion** Continuous assessment Presential Duration: 02:00
4	**Event-B and related topics** Duration: 02:00 **Quizzes** Duration: 01:00			
5	**Event-B and related topics** Duration: 02:00 **Event-B and related topics** Duration: 02:00			
6	**Event-B and related topics** Duration: 01:00			**Homework: solutions and discussion** Continuous assessment Presential Duration: 02:00
7	**Event-B and related topics** Duration: 02:00 **Quizzes** Duration: 01:00			

8	**Event-B and related topics** Duration: 02:00 **Quizzes** Duration: 01:00			
9	**Event-B and related topics** Duration: 01:00 **Event-B and related topics** Duration: 02:00			**Homework: solutions and discussion** Continuous assessment Presential Duration: 02:00
10	**Quizzes** Duration: 01:00 **Event-B and related topics** Duration: 02:00			
11	**Event-B and related topics** Duration: 02:00 **Quizzes** Duration: 01:00			
12	**Quizzes** Duration: 01:00 **Logic-based programming languages** Duration: 02:00			
13	**Logic-based programming languages** Duration: 02:00			**Homework: solutions and discussion** Continuous assessment Presential Duration: 01:00
14	**Quizzes** Duration: 01:00 **Logic-based programming languages** Duration: 02:00			
15	**Logic-based programming languages** Duration: 02:00			**Homework: solutions and discussion** Continuous assessment Presential Duration: 01:00
16				

17				**Presentation of a development made with one of the tools studied in the course** Continuous assessment Presential Duration: 03:00 **Final regular exam** Final examination Presential Duration: 03:00

Depending on the programme study plan, total values will be calculated according to the ECTS credit unit as 26/27 hours of student face-to-face contact and independent study time.

* The schedule is based on an a priori planning of the subject; it might be modified during the academic year, especially considering the COVID19 evolution.

7. Activities and assessment criteria

7.1. Assessment activities

7.1.1. Continuous assessment

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
3	Homework: solutions and discussion		Face-to-face	02:00	14%	0 / 10	CE07 CE03
6	Homework: solutions and discussion		Face-to-face	02:00	14%	0 / 10	CE07 CE03
9	Homework: solutions and discussion		Face-to-face	02:00	14%	0 / 10	CG07 CE07 CE03
13	Homework: solutions and discussion		Face-to-face	01:00	9%	0 / 10	CG07 CE07 CE03
15	Homework: solutions and discussion		Face-to-face	01:00	9%	0 / 10	CG07 CE07 CE03
17	Presentation of a development made with one of the tools studied in the course		Face-to-face	03:00	40%	5 / 10	CG07 CE07 CE03

7.1.2. Final examination

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
17	Final regular exam		Face-to-face	03:00	100%	5 / 10	CG07 CE07 CE03

7.1.3. Referred (re-sit) examination

Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
Extra final exam		Face-to-face	03:00	100%	5 / 10	CE07 CE03 CG07

7.2. Assessment criteria

Students will be evaluated based on their performance in the course homework / quizzes and the project. In the presentation, the quality of the information and the ability to answer questions on the decision designs will be taken into account. All students participating in a project are expected to also present part of the project and be able to answer questions to any part of the project.

8. Teaching resources

8.1. Teaching resources for the subject

Name	Type	Notes
Lawrence Paulson's class notes	Bibliography	Lawrence Paulson?s Logic and Proof are the course notes of the author for a Logic course in Cambridge. Highly recommended, as they are both rigorous and very concise. They provide very good background material for both parts of the course.
Logic in Computer Science (Huth and Ryan)	Bibliography	A very good book on the use of logic in computer science is Logic in Computer Science, by Huth and Ryan. The Computer Science School should have several copies. There may be electronic copies on the Internet, if possible of the second edition.
http://wiki.event-b.org/	Web resource	Central Event-B site
Modeling in Event-B: System and Software Engineering, by Jean-Raymond Abrial.	Bibliography	The reference book for Event B, with plenty of worked examples.
http://ciao-lang.org/index.html	Web resource	Web site of the Ciao system
An overview of Ciao and its design philosophy	Bibliography	A paper describing the design principles behind Ciao Prolog: http://cliplab.org/papers/hermenegildo11:ciao-design-tplp.pdf

9. Other information

9.1. Other information about the subject

This course will be given in English. Please note that in case Spanish appears as the course language in the general description, that would be a clerical mistake.

It is expected that the health situation for the Spring semester would have improved enough as to make it possible to use fully the classrooms. Therefore, face-to-face teaching has been planned.

If the health situation does not allow fully using the classrooms, teaching will change to a mixed online / face-to-face model.