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

2023/24 - 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	2023-24

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

CB07 - Que los estudiantes sepan aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios (o multidisciplinares) relacionados con su área de estudio

CB10 - Que los estudiantes posean las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.

CE01 - Capacidad para expresar los requisitos y propiedades que ha de satisfacer un sistema informático, en una variedad de lenguajes formales y a diferentes niveles de detalle.

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.

4.2. Learning outcomes

RA10 - LO4 - Ability to deduce verifying conditions that programs must satisfy.

RA1 - Acquaintance with various techniques for formal software development

RA3 - Knowledge of techniques for proving code correctness

RA12 - Ability to specify safety and liveness properties of distributed systems

RA14 - Ability to competently use tools for analyzing and validating distributed systems

RA5 - Effective use of rigorous software development techniques

* 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

6. Schedule

6.1. Subject schedule*

Week	Classroom activities	Laboratory activities	Distant / On-line	Assessment activities
1	**Introduction to formal methods and correctness by construction** Duration: 01:30 Lecture **Sample cases of formal development** Duration: 01:30 Cooperative activities			
2	**Event-B and related topics** Duration: 02:00 Lecture **Quizzes** Duration: 01:00 Problem-solving class			
3	**Event-B and related topics** Duration: 02:00 Lecture **Quizzes** Duration: 01:00 Problem-solving class			**Homework** Individual work Continuous assessment Not Presential Duration: 04:00
4	**Event-B and related topics** Duration: 02:00 Lecture **Quizzes** Duration: 01:00 Problem-solving class			
5	**Event-B and related topics** Duration: 02:00 Lecture **Quizzes** Duration: 01:00 Problem-solving class			
6	**Event-B and related topics** Duration: 02:00 Lecture **Quizzes** Duration: 01:00 Problem-solving class			**Homework** Individual work Continuous assessment Not Presential Duration: 04:00

7	**Event-B and related topics** Duration: 02:00 Lecture **Quizzes** Duration: 01:00 Problem-solving class			
8	**Event-B and related topics** Duration: 02:00 Lecture **Quizzes** Duration: 01:00 Problem-solving class			
9	**Event-B and related topics** Duration: 02:00 Lecture **Quizzes** Duration: 01:00 Problem-solving class			**Homework** Individual work Continuous assessment Not Presential Duration: 08:00
10	**Quizzes** Duration: 01:00 Problem-solving class **Event-B and related topics** Duration: 02:00 Lecture			
11	**Presentation of term project** Duration: 01:00 Additional activities **Event-B and related topics** Duration: 02:00 Lecture			**Term project** Group work Continuous assessment Not Presential Duration: 20:00
12	**Event-B and related topics** Duration: 02:00 Lecture **Quizzes** Duration: 01:00 Problem-solving class			
13	**Event-B and related topics** Duration: 02:00 Lecture **Quizzes** Duration: 01:00 Problem-solving class			
14	**Event-B and related topics** Duration: 02:00 Lecture **Quizzes** Duration: 01:00 Problem-solving class			

15				**Homework** Group presentation Continuous assessment Presential Duration: 03:00
16				
17				**Final regular exam** Written test 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. Assessment

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
3	Homework	Individual work	No Presential	04:00	20%	2 / 10	CB07 CB10 CE01 CE03
6	Homework	Individual work	No Presential	04:00	20%	2 / 10	CB07 CB10 CE01 CE03
9	Homework	Individual work	No Presential	08:00	20%	2 / 10	CB07 CB10 CE01 CE03
11	Term project	Group work	No Presential	20:00	40%	4 / 10	CB07 CB10 CE01 CE03
15	Homework	Group presentation	Face-to-face	03:00	%	4 / 10	CB07 CB10 CE01 CE03

7.1.2. Global examination

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
17	Final regular exam	Written test	Face-to-face	03:00	100%	5 / 10	CB07 CB10 CE01 CE03

7.1.3. Referred (re-sit) examination

Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
Extra final exam	Written test	Face-to-face	03:00	100%	5 / 10	CB07 CB10 CE01 CE03

7.2. Assessment criteria

- No mandatory activities are necessary to pass via the final exams
- The minimum grade to pass the course is 5 over 10 (either when it is calculated as the weighted sum of individual homework or when it is the grade of a single comprehensive exam).
- The topics covered in the different homework assignments cannot be tested separately in the final exam, as they are deeply intertwined and are not isolated units of knowledge.
- The global exams, both the regular and the extraordinary ones, will be in person.

- Copying from any source (either textbooks, the Internet, another student, or any other source) with or without the permission of the author of the source, as well as other types of academic fraud, can lead to a 'fail' grade in the course and / or being reported to the academic authorities, who will decide whether to take additional authoritative measures. In particular, in case of non-ethical or fraudulent behavior, the Law 3/2022 of February 24th will be applied, as well as the corresponding UPM regulations. Article 12 and 14 of Law 3/2022 states that a serious fault may mean, among other outcomes, failing the corresponding sitting.
- There are no learning blocks whose earned grades can be carried over to future academic courses.
- Failure to deliver a homework assignment at the time and in the form stated by the instructor(s) may result in a failure for that exercise.
- For progressive evaluation: if for any reason it is not possible to prepare / hand out some homework assignment, its weight in the final grade will be split among the rest of the homework exercises in such a way that the relative weight of the rest of the assignments, when compared with each other, will be the same they had before removing the homework that could not be handed out.

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.

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.