# ANX-PR/CL/001-01

# LEARNING GUIDE

## SUBJECT

**103000657 - Correctness By Construction**

## DEGREE PROGRAMME

10AM - Master Universitario En Ingenieria Del Software

## ACADEMIC YEAR & SEMESTER

2022/23 - Semester 2

# Index

## Learning guide

# 1. Description

## 1.1. Subject details

| | |
|---|---|
| **Name of the subject** | 103000657 - 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** | 10AM - Master Universitario en Ingenieria del Software |
| **Centre** | 10 - Escuela Tecnica Superior De Ingenieros Informaticos |
| **Academic year** | 2022-23 |

# 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

CE13 - Tener una visión de los distintos aspectos específicos y emergentes de la ingeniería del software, y profundizar en algunos de ellos

CE14 - Comprender lo que pueden y no pueden conseguir las prácticas actuales de ingeniería del software, y sus limitaciones y su posible futura evolución.

CG13 - Apreciación de los límites del conocimiento actual y de la aplicación práctica de la tecnología más reciente

### 4.2. Learning outcomes

RA25 - Communication skills in public SC13, SC14, CG3, CG18 S

RA26 - Group work skill SC13, SC14, CG17 A

RA50 - Posee las técnicas necesarias para la realización de un informe o memoria sobre un trabajo realizado en un entorno socio?lingüístico nacional/internacional.

RA66 - RA-AV-2 Acquaintance with various techniques for formal software development

RA69 - RA-AV-4 - Knowledge of techniques for formally proving code correctness.

RA65 - RA-AV-1 - Acquaintance with design requirements and implementation requirements.

RA54 - Proponer una solución justificada a un problema real que sea complejo o mal definido, o perteneciente a un área nueva o emergente, o que requiera el desarrollo de enfoques o métodos nuevos y originales, dentro del contexto de la ingeniería del software justificándola de una forma cualitativa y cuantitativa.

RA68 - RA-AV-3 - Knowledge of languages for formal specification

RA91 - Apply techniques for modelling the context of use

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

| | | | | |
|---|---|---|---|---|
| 7 | **Event-B and related topics**<br>Duration: 02:00<br>Lecture<br><br>**Quizzes**<br>Duration: 01:00<br>Problem-solving class | | | |
| 8 | **Event-B and related topics**<br>Duration: 02:00<br>Lecture<br><br>**Quizzes**<br>Duration: 01:00<br>Problem-solving class | | | |
| 9 | **Event-B and related topics**<br>Duration: 02:00<br>Lecture<br><br>**Discussion of Homerwork**<br>Duration: 01:00<br>Problem-solving class | | | **Homework**<br>Individual work<br>Continuous assessment<br>Not Presential<br>Duration: 08:00 |
| 10 | **Quizzes**<br>Duration: 01:00<br>Problem-solving class<br><br>**Event-B and related topics**<br>Duration: 02:00<br>Lecture | | | |
| 11 | **Presentation of term project**<br>Duration: 03:00<br>Additional activities | | | **Term project**<br>Group work<br>Continuous assessment<br>Not Presential<br>Duration: 10:00 |
| 12 | **Logic-based programming languages**<br>Duration: 03:00<br>Lecture | | | |
| 13 | **Logic-based programming languages**<br>Duration: 03:00<br>Lecture | | | **Homework**<br>Individual work<br>Continuous assessment<br>Not Presential<br>Duration: 03:00 |
| 14 | **Logic-based programming languages**<br>Duration: 03:00<br>Lecture | | | |
| 15 | **Logic-based programming languages**<br>Duration: 03:00<br>Lecture | | | **Homework**<br>Individual work<br>Continuous assessment<br>Not Presential<br>Duration: 03:00 |
| 16 | | | | |
| 17 | | | | **Final regular exam**<br>Written test<br>Final examination<br>Presential<br>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 | 15% | 0 / 10 | CG13<br>CE14 |
| 6 | Homework | Individual work | No Presential | 04:00 | 15% | 0 / 10 | CE13 |
| 9 | Homework | Individual work | No Presential | 08:00 | 15% | 2 / 10 | CE13 |
| 11 | Term project | Group work | No Presential | 10:00 | 40% | 4 / 10 | CE13 |
| 13 | Homework | Individual work | No Presential | 03:00 | 7.5% | 0 / 10 | CE13 |
| 15 | Homework | Individual work | No Presential | 03:00 | 7.5% | 0 / 10 | CE13 |

### 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 | CG13<br>CE14<br>CE13 |

### 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 | CG13<br>CE14<br>CE13 |

## 7.2. Assessment criteria

- No mandatory activities necessary are to pass via the global 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.
- For the same reason, there is not a minimum grade in the initial homework assignments, as they can be compensated by later assignments.
- 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.
- Active participation in the course can be taken into account to fine-tune the student's final grade.
- 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. |
| 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 |
| wp.software.imdea.org/cbc | Web resource | Web site of the Event-B part of the course |

# 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.