ANX-PR/CL/001-01

LEARNING GUIDE

SUBJECT

103000806 - Design Of Correct-by-construction Systems

DEGREE PROGRAMME

10AR - Master Interuniversitario En Métodos Formales En Ingeniería Informática

ACADEMIC YEAR & SEMESTER

2020/21 - Semester 2

Index

Learning guide

1. Description

1.1. Subject details

Name of the subject	103000806 - Design Of Correct-By-Construction Systems
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	2020-21

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.
Angel Herranz Nieva	2309	angel.herranz@upm.es	Tu - 09:00 - 10:00 W - 09:00 - 10:00 Th - 09:00 - 12:00 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

CB06 - Poseer y comprender conocimientos que aporten una base u oportunidad de ser originales en el desarrollo y/o aplicación de ideas, a menudo en un contexto de investigación

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.

CE02 - Capacidad para utilizar de forma competente las herramientas existentes de demostración automática y asistida de teoremas y de propiedades matemáticas.

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.

CE05 - Capacidad para utilizar y desarrollar herramientas que analizan propiedades de los programas, mediante su ejecución en un conjunto de casos cuidadosamente seleccionado.

CE06 - Capacidad para utilizar modelos de cómputo alternativos a los convencionales, tales como la computación cuántica, los sistemas de reescritura, las máquinas abstractas, la programación con restricciones o los algoritmos bio-inspirados.

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.

CE08 - Capacidad para la dirección de proyectos de investigación, desarrollo e innovación, en empresas y centros tecnológicos, con garantía de la seguridad para las personas y bienes, y la calidad final de los productos.

CE10 - Capacidad para la modelización matemática, el cálculo y la simulación en centros tecnológicos y de ingeniería de empresa, particularmente en tareas de investigación, desarrollo e innovación en todos los ámbitos

relacionados con la Ingeniería en Informática.

CG01 - Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en todos los ámbitos de la Ingeniería Informática.

CG05 - Capacidad para la aplicación de los conocimientos adquiridos para resolver problemas en entornos nuevos o poco conocidos dentro de contextos amplios y multidisciplinares, siendo capaces de integrar dichos conocimientos.

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

CT01 - Capacidad para trabajar en equipo, ya sea como un miembro más o realizando la labor de dirección del mismo, promoviendo el libre intercambio de ideas.

CT02 - Capacidad para fomentar la creatividad tanto propia como la de los restantes miembros del equipo.

CT03 - Capacidad de razonamiento crítico como vía para mejorar la generación y desarrollo de ideas en un contexto profesional o de investigación.

4.2. Learning outcomes

RA1 - Acquaintance with various techniques for formal software development

RA2 - Knowledge of languages which ease the application of the aforementioned techniques.

RA5 - Effective use of rigorous software development techniques

RA3 - Knowledge of techniques for proving code correctness

RA4 - Acquaintance with design requirements and implementation requirements

* 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 in Practice
2. Event-B Basics
3. First-Order Logic and Sequent Calculus
4. A First Example in Event-B
5. The Rodin Tool
6. Sequential Systems. Refinement.
7. Concurrent Systems. Refinement.
8. Other Topics:
   - 8.1. Analysis-Based Development Environments.
   - 8.2. Dependent Types & Verification.
   - 8.3. Calculus of Constructions, Coq.

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: 02:00 **Quizzes** Duration: 01:00			
4	**Event-B and related topics** Duration: 02:00 **Quizzes** Duration: 01:00			
5				**Homework: solutions and discussion** Continuous assessment Presential Duration: 03:00
6	**Event-B and related topics** Duration: 02:00 **Quizzes** Duration: 01:00			
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				**Homework: solutions and discussion** Continuous assessment Presential Duration: 03:00
10	**Quizzes** Duration: 01:00 **Calculus of Constructions (planned)** Duration: 02:00			
11	**Analysis-Based Development Environments (planned)** Duration: 02:00 **Quizzes** Duration: 01:00			
12	**Analysis-Based Development Environments (planned)** Duration: 02:00 **Quizzes** Duration: 01:00			
13	**Dependent types (planned)** Duration: 02:00 **Quizzes** Duration: 01:00			
14	**Dependent types (planned)** Duration: 02:00 **Quizzes** Duration: 01:00			
15				**Project presentation** Continuous assessment Presential Duration: 03:00

16				
17				**Presentation of a development made with one of the tools studied in the course** 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
5	Homework: solutions and discussion		Face-to-face	03:00	20%	0 / 10	CT01 CE01 CE03 CE05 CE06 CE07 CE10 CG07 CT02 CT03 CB07 CE02 CE08 CG01 CG05 CB06 CB10
9	Homework: solutions and discussion		Face-to-face	03:00	20%	0 / 10	CT03 CB07 CT01 CE01 CE03 CE05 CE06 CE07 CE10 CG07 CT02 CE02 CE08 CG01 CG05 CB06 CB10

15	Project presentation		Face-to-face	03:00	60%	0 / 10	CT03 CB07 CT01 CE01 CE03 CE05 CE06 CE07 CE10 CG07 CT02 CE02 CE08 CG01 CG05 CB06 CB10

7.1.2. Final examination

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
17	Presentation of a development made with one of the tools studied in the course		Face-to-face	03:00	100%	5 / 10	CB07 CT01 CE01 CE03 CE05 CE06 CE07 CE10 CT03 CG07 CT02 CE02 CE08 CG01 CG05 CB06 CB10

7.1.3. Referred (re-sit) examination

Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
Course exam		Face-to-face	03:00	100%	5 / 10	CT03 CB07 CT01 CE01 CE03 CE05 CE06 CE07 CE10 CG07 CT02 CE02 CE08 CG01 CG05 CB06 CB10

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
Event B development environment	Others	
Modeling in Event-B: System and Software Engineering. Jean-Raymond Abrial. Cambridge University Press.	Bibliography	
http://wiki.event-b.org/	Bibliography	
Seven Myths of Formal Methods. Anthony Hall. IEEE Software, September 1990	Bibliography	
Seven More Myths of Formal Methods. Jonathan P. Bowen, Michael G. Hinchey. IEEE Software, July 1995.	Bibliography	
First Steps in the Verified Software Grand Challenge. Cliff Jones, Peter O'Hearn, Jim Woodcock. IEEE Computer, April 2006.	Bibliography	
Coq	Others	
Ciao/CiaoPP	Others	

Liquid Haskell	Others	

9. Adendas

- Lectures may take place online if deemed necessary due to the public health situation at the time the semester starts. In that case, students will be given information on how to access the online platform ahead of time.