

=

European Software
Skills Alliance.

ESSA Learning programmes

ANNEX I Junior Developer EQF 4/5

30 November 2024
Status: Final version

ESSA

Copyright © 2024 by the European Software Skills Alliance. The project resources contained herein are publicly available under the [Creative Commons license 4.0 B.Y](https://creativecommons.org/licenses/by/4.0/)

ESSA Learning programme – Junior Developer EQF 4/5, 2024.

Deliverable 10 – ESSA Learning Programmes & Materials – ANNEX I

This document is a draft version and is subject to change after review coordinated by the European Education and Culture Executive Agency (EACEA).

Author: Adecco/Mylia

softwareskills.eu

Legal Disclaimer

The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



Co-funded by the
Erasmus+ Programme
of the European Union

About ESSA

The European Software Skills Alliance (ESSA) is a four-year transnational project funded under the EU's Erasmus+ programme. It ensures the skills needs of the rapidly evolving Software sector can be met — today and tomorrow.

ESSA provides current and future software professionals, learning providers and organisations with software needs with the educational and training instruments they need to meet the demand for software skills in Europe.

ESSA will develop a European Software Skills Strategy and learning programmes for Europe. It will address skill mismatches and shortages by analysing the sector in depth and delivering future-proof curricula and mobility solutions; tailored to the European software sector's reality and needs.

Project partners

The ESSA consortium is led by DIGITALEUROPE. It is composed of academic and non-academic partners from the education, training, and software sectors.

View all project partners: [ESSA Partners](#) | [ESSA Associated Partners](#)



Table of Contents

1	Junior Developer EQF 4/5 – ESSA Learning Programme	6
1.1	Students with ICT background	6
1.1.1	PLO 1. Application Design [e-2]	6
1.1.2	Learning Resources - PLO 1. Application Design [e-2]	9
1.1.3	PLO 2. Application Development [e-2]	11
1.1.4	Learning Resources – PLO 2. Application Development [e-2]	14
1.1.5	PLO 3. Component Integration [e-2]	18
1.1.6	Learning Resources – PLO 3. Component Integration [e-2]	20
1.1.7	PLO 4. Testing [e-2]	25
1.1.8	Learning Resources – PLO 4. Testing [e-2]	27
1.1.9	PLO 5. Documentation Production [e-2]	28
1.1.10	Learning Resources – PLO 5. Documentation Production [e-2]	30
1.1.11	PLO 6. Problem management [e-2]	32
1.1.12	Learning Resources – PLO 6. Problem Management [e-2]	34
1.1.13	PLO 7. Professional related competences [EQF5]	35
1.1.14	Learning Resources – PLO 7. Profession related competence [EQF5]	37
1.1.15	PLO 8. Soft competences [EQF5]	39
1.1.16	Learning Resources – PLO 8. Soft competences [EQF5]	42
1.1.17	PLO 9. Functioning in organisations [EQF5]	44
1.1.18	Learning Resources – 9. PLO Functioning in organisation [EQF5]	46

List of abbreviations and acronyms

Abbreviation	Term
e-CF, EN 16234-1	European e-Competence Framework, European Norm 16234 - Part 1: Framework
ECTS	European Credit Transfer and Accumulation System
EQF	European Qualifications Framework
ESSA	European Software Skills Alliance
LO	Learning Outcome
PLO	Programme Learning Outcome

1 Junior Developer EQF 4/5 – ESSA Learning Programme

1.1 Students with ICT background

Executive summary

The Learning programme is being designed by Adecco Formazione (Myliia) (IT) to develop the skills of students from Technical Institutes, University students and professionals committed in upskilling or reskilling paths with a prior basic knowledge of the topic.

The Learning programme provides participants with the knowledge necessary for software development activity at junior level. Developer-related skills such as: programming languages, techniques and practices for frontend and backend development, the main support tools, the fundamentals of software testing, project management techniques and team collaboration skills are explored to ensure maximum productivity in business contexts.

Targeted Institutions: Higher Education and VET providers.

The recommended Learning programme is articulated in thirty-four (34) Learning Units, for a total of 404 hours of study and 16,16 ECTS.

The recommended delivery method is the Virtual Classroom.

1.1.1 PLO 1. Application Design [e-2]¹

1. PLO Application Design [e-2]

The learner has demonstrated capability

→ to interpret a design for a software application or component

Unit learning outcomes	Explains and distinguishes basic principles and terminology of software design (e.g., phases in the design process, common techniques, deliverables)
	Describes principles of user interface design
	Reads design models and diagrams (e.g., ERD, UML)
	Interprets a basic database design
	Interprets a design for an application or software component

¹ Unit Learning outcomes are directly sourced from the ESSA Educational Profiles developed according to different EQF levels in previous WP3. For full consultation, refer to <https://softwareskills.eu/library/essa-educational-profiles-for-software-roles/>

1.1.1.1 Duration of Study

Recommended duration: starting from n.0,66 ECTS

Often integrated with studies of PLOs: PLO 2 Application Development [e-2]

1.1.1.2 Recommendations for Micro-credentials

N/A

1.1.1.3 Recommendations on Didactical Approach, Delivery Methods and Training Environment

Recommended didactical approach:

- Virtual Classroom
- Work placement

Recommended delivery methods:

- Lecture up to 60%
- Coding Training Lab delivered by individual/team project work up to 40%

Additional comments

It is recommended to deepen the topics presented in the Learning Units by reading publications dedicated to the various topics, reading websites specialized in programming, watching online tutorials and downloading materials useful for practical exercises from official sources.

1.1.1.4 WBL and Follow-up Reinforcement

After learning the basic principles, terminology, and models of software design, the study should focus on analysing and simulating real work-life-like tasks as, for example the student:

- Specifies a design for an application or software (component), taking into account certain basic constraints/ requirements;
- Checks whether the design meets requirements/ wishes and, if necessary, makes proposals to adapt the design;
- Interprets and employs the software (component)/ application design and design patterns.

1.1.1.5 Important (new) approaches and technologies to consider

Before starts to coding, the learner should decide if he/she wants to start off with a set of codes and stick with them (deductive coding), or come up with the codes as he/she read what he/she read see in his/her data (inductive), or take a combination approach.

1.1.1.6 Assessment

Unit learning outcome	Assessment method	Validation of prior acquired competences (skills and knowledge)
Collects, formalises and validates functional and non-functional requirements;	Exam: The candidate must evaluate which programming language is most suitable for the development of specific digital services.	n/a
Checks whether the design meets requirements/ wishes and, if necessary, makes proposals to adapt the design	Exam: The candidate must evaluate which programming language is most suitable for the development of specific digital services.	n/a
Specifies a design for an application or software (component), taking into account certain basic constraints/ requirements	Exam: The candidate must evaluate which programming language is most suitable for the development of specific digital services.	n/a
Interprets and employs the software (component)/ application design and design patterns.	Exam: The candidate must evaluate which programming language is most suitable for the development of specific digital services.	n/a
Collects, formalises and validates functional and non-functional requirements	Assignment: Practical activity. The learner is asked to build the best User Experience for the web application developed in the previous modules.	n/a
Drafts functional and technical design	Assignment: Practical activity. The learner is asked to build the best User Experience for the web application developed in the previous modules.	n/a
Specifies a design for an application or software (component), taking into account certain basic constraints/ requirements	Assignment: Practical activity. The learner is asked to build the best User Experience for the web application developed in the previous modules.	n/a
Checks whether the design meets requirements/ wishes and, if necessary, makes proposals to adapt the design.	Assignment: Practical activity. The learner is asked to build the best User Experience for the web application developed in the previous modules.	n/a

1.1.2 Learning Resources - PLO 1. Application Design [e-2]

LEARNING UNIT	EQF	Duration	Didactical Approach	ASSESSMENT	TOPIC	Delivery method of the learning material	Learning materials
1.1_Overview of the Main Programming Languages and key differences	5	8 hours	The didactic approach would be aimed to allows participants to understand the characteristics and differences between the main programming languages, using practical examples, such as viewing and analyzing programming code.	1 exam. The candidate must evaluate which programming language is most suitable for the development of specific digital services.	Overview of the Main Programming Languages and key differences ISTQB Syllabus lecture notes ISTQB CTFL 4 Sample exam questions lecture notes	Virtual classroom, Workshop and lecture guides	LU 1.1_Overview of the main programming languages_lesson.pptx LU 1.1_ISTQB_CTFL_Syllabus-v4.0_Lesson_Lecture notes.pdf LU 1.1_ISTQB_CTFL_v4.0_Sample-Exam-A-Questions_v1.6_Lecture notes.pdf
1.2_Principles of UI/UX Design. Adobe XD, Zeplimg.	5	12 hours	The didactic approach would be aimed to allow participants to understand the main UI/UX design principles, using practical	Assignment: practical activity. The student is asked to build the best User Experience for the web application developed in the	Principles of UI UX Design	Virtual classroom, Workshop and lecture guides	LU 1.2_Principles of UI-UX Design_lesson.pptx

	examples, such as viewing and analyzing programming code.	previous modules.		
--	---	-------------------	--	--

1.1.3 PLO 2. Application Development [e-2]²

2. PLO Application Development [e-2]

The learner has demonstrated capability

→ to systematically develop a simple software application or component

→ to propose modifications to an existing solution

→ to document the development activities

Unit learning outcomes	Explains and distinguishes common software development methods (e.g., waterfall, iterative, agile), techniques (e.g., object-oriented) and tools (e.g., IDE, CASE; multimedia integration tools; app development tools)
	Describes common programming principles and terminology (e.g., secure programming)
	Explains concepts and principles of databases, data structures and query languages (e.g., SQL)
	Participates in a development process and applies a common software development method (e.g., agile)
	Creates a simple relational database
	Writes code and related documentation to it, by using a common programming language and applying coding conventions (e.g., Java, Javascript, PHP, Python; clean coding principle)
	Creates a simple working software component or application, taking into account architecture, design requirements and other possible constraints (e.g., installability) applying relevant tools and techniques (e.g., object-oriented programming; IDE, CASE; editors, compilers, version control tools)
	Modifies an existing software application or component

1.1.3.1 Duration of Study

Recommended duration: starting from n. 4 ECTS

Often integrated with studies of PLOs: 1- 3-4-5

1.1.3.2 Recommendations for Micro-credentials

N/A

1.1.3.3 Recommendations on Didactical Approach, Delivery Methods and Training Environment

Recommended didactical approach:

² Unit Learning outcomes are directly sourced from the ESSA Educational Profiles developed according to different EQF levels in previous WP3. For full consultation, refer to <https://softwareskills.eu/library/essa-educational-profiles-for-software-roles/>

- Virtual Classroom

Additional comments

n/a

Recommended delivery methods:

- Lecture up to 60%
- Coding Training Lab delivered by individual/team project work up to 40%

Additional comments

It is recommended to deepen the topics presented in the Learning Units by reading publications dedicated to the various topics, reading websites specialized in programming, watching online tutorials and downloading materials useful for practical exercises from reliable sources.

1.1.3.4 WBL and Follow-up Reinforcement

After learning the basic principles, terminology, and models of Application Development, the study should focus on analysing and simulating real work-life-like tasks as, for example, the student:

- Participates in a development process and applies a common software development method;
- Writes code and related documentation, by using a common programming language and applying coding conventions (e.g., Java, Javascript, PHP, Python; clean coding principle)

1.1.3.5 Important (new) approaches and technologies to consider

N/A

1.1.3.6 Assessment

Unit learning outcome	Assessment method	Validation of prior acquired competences (skills and knowledge)
Explains and distinguishes common software development methods (e.g., waterfall, iterative, agile), techniques (e.g., object-oriented) and tools (e.g., IDE, CASE; multimedia integration tools; app development tools)	1 exam. The candidate must evaluate which programming language is most suitable for the development of specific digital services.	n/a
Describes common programming principles and terminology (e.g., secure programming)	1 exam. The candidate must evaluate which programming language is most suitable for the development of specific digital services.	n/a
Explains concepts and principles of databases, data structures, and query languages (e.g., SQL)	1 exam. The candidate is asked to set up a database to support a web application	n/a

Participates in a development process and applies a common software development method (e.g., agile)	1 exam. The candidate must evaluate which programming language is most suitable for the development of specific digital services.	n/a
Creates a simple relational database	1 exam. The candidate is asked to set up a database to support a web application	n/a
Writes code and related documentation to it, by using a common programming language and applying coding conventions	Assignment: practical activity. The student is asked to create a web application using HTML5, CSS3 and Bootstrap.	n/a
Creates a simple working software component or application, taking into account architecture, design requirements and possible other constraints (e.g., installability) applying relevant tools and techniques	Assignment: practical activity. The student is asked to create a web application using HTML5, CSS3 and Bootstrap.	n/a
Modifies an existing software application or component	Assignment: practical activity. The student is asked to modify a web application using HTML5, CSS3 and Bootstrap.	n/a

1.1.4 Learning Resources – PLO 2. Application Development [e-2]

LEARNING UNIT	EQF	Duration	Didactical Approach	ASSESSMENT	TOPIC	Delivery method of the learning material	Learning materials
2.1 Overview of the Main Programming Languages and key differences;	5	8 hours	The didactic approach would be aimed to allows participants to understand the main programming languages and their applications using practical examples, such as viewing and analyzing programming code.	1 exam. The candidate must evaluate which programming language is most suitable for the development of specific digital services.	Overview of the Main Programming Languages and key differences	Workshop and lecture guides	LU 2.1_Overview of the main programming languages_Lesson.pptx
2.2 HTML5, CSS3, BOOTSTRAP	5	24 hours	The didactic approach would be aimed to allows participants to understand the HTML and CSS code and	Assignment: practical activity. The student is asked to create a web application using HTML5,	HTML5, CSS3, Bootstrap	Workshop and lecture guides	LU 2.2_HTML5 CSS3 Bootstrap_Lesson.pptx

			relative application, using practical examples, such as viewing and analyzing programming code.	CSS3 and Bootstrap.			
2.3 Javascript, AJAX, Typescript, GIT	5	24 hours	The didactic approach would be aimed to allows participants to understand the main applications of javascript, ajax, typescript, using practical examples, such as viewing and analyzing programming code.	Assignment: practical activity. The student is asked to develop a web application using frontend coding (Javascript, AJAX, Typescript) and related tools.	Java at main Javascript Ajax	Workshop and lecture guides	LU 2.3_Java at main_Lesson.pptx LU 2.3_Javascript Ajax_Lesson.pptx
2.4 Backend – Coding and development tools: Java 11, Spring Boot, Sprint Data, Hibernate, Ex	5	16 hours	The didactic approach would be aimed to allows participants to understand the main	Assignment: practical activity. The student is asked to build and test a backend-side	Backend – Java 11	Workshop and lecture guides	LU 2.4_Backeng_Java 11_Lesson.pptx

Java: Junit, Mockito.			applications of those sets of code using practical examples, such as viewing and analyzing programming code.	application using Java 11 and related development tools.			
2.5 Backend – Coding and development tools: PHP, Laravel, Eloquent	5	16 hours	The didactic approach would be aimed to allows participants to understand the main applications of those sets of code using practical examples, such as viewing and analyzing programming code.	Assignment: practical activity. The student is asked to build and test a backend-side application using PHP and related development tools.	PHP Laravel Eloquent	Workshop and lecture guides	LU 2.5_PHP Eloquent_Lesson.pptx Laravel
2.6 Backend – Coding and development tools: Fundamentals of Ruby,	5	16 hours	The didactic approach would be aimed to allows participants to understand the	Assignment: practical activity. The student is asked to build a backend-	Back End – Objects – Ruby – Phyton -NodeJS	Workshop and lecture guides	LU 2.6_Back End – Objects – Ruby – Phyton -NodeJS_Lesson.pptx

Python, NodeJS			main applications of those sets of code using practical examples, such as viewing and analyzing programming code.	side application using Ruby, Python and NodeJS.			
2.7 Agile Project Management, SCRUM and collaboration tools	5	8 hours	The didactic approach would be aimed to allows participants to understand the Agile and SCRUM culture and framework using practical examples and exercises.	Assignment: practical activity. The student is invited to apply the Agile methodology in the development of a web application.	Agile PM and SCRUM	Workshop and lecture guides	LU 2.7_Agile PM and SCRUM_Lesson.pptx

1.1.5 PLO 3. Component Integration [e-2]³

3. PLO Component Integration [e-2]

The learner has demonstrated capability

→ to integrate efficiently a software application or component into an existing system

→ to document the installation activities

Unit learning outcomes	Explains and distinguishes common methods, techniques and tools related to efficient integration
	Describes the interplay between and compatibility of system components
	Carries out installation and configuration activities, applying common methods, techniques and tools related to efficient integration (e.g., packaging and distribution, virtualisation, containerisation)
	Monitors and tests the connectivity of integrated systems
	Writes an installation report

1.1.5.1 Duration of Study

Recommended duration: starting from n.4 ECTS

Often integrated with studies of PLOs: 1 - 5

1.1.5.2 Recommendations for Micro-credentials

N/A

1.1.5.3 Recommendations on Didactical Approach, Delivery Methods and Training Environment

Recommended didactical approach:

- Virtual Classroom

Additional comments

It is recommended to deepen the topics presented in the Learning Units by reading publications dedicated to the various topics, reading websites specialized in programming, watching online tutorials and downloading materials useful for practical exercises from reliable sources.

³ Unit Learning outcomes are directly sourced from the ESSA Educational Profiles developed according to different EQF levels in previous WP3. For full consultation, refer to <https://softwareskills.eu/library/essa-educational-profiles-for-software-roles/>

Recommended delivery methods

- Lecture up to 60%
- Coding Training Lab delivered by individual/team project work up to 40%

1.1.5.4 WBL and Follow-up Reinforcement

After learning the basic principles, terminology, and models of Component Integration, the study should focus on analysing and simulating real work-life-like tasks as, for example, the student:

- Participates in a development process and applies a common software development method;
- Writes code and related documentation, by using a common programming language and applying coding conventions;
- Describes the interplay between and compatibility of system components;
- Monitors and tests the connectivity of integrated systems.

1.1.5.5 Important (new) approaches and technologies to consider

Not Applicable

1.1.5.6 Assessment

Unit learning outcome	Assessment method	Validation of prior acquired competences (skills and knowledge)
Creates basic database design	1 exam. The candidate is asked to set up a database to support a web application	n/a
Creates a working software component/ application by applying design requirements and eventually other constraints/ requirements;	Assignment: practical activity. The student is asked to create a web application using HTML5, CSS3 and Bootstrap.	n/a
Modifies an existing software component/ application, in order to optimize application	The student is asked to optimize a web application using frontend coding and related tools.	n/a

1.1.6 Learning Resources – PLO 3. Component Integration [e-2]

LEARNING UNIT	EQF	Duration	Didactical Approach	ASSESSMENT	TOPIC	Delivery method of the learning material	Learning materials
3.1 Entity-Relationship Model, SQL, MySql	5	8 hours	The didactic approach would be aimed to allows participants to understand the main applications of those sets of code using practical examples, such as viewing and analyzing programming code.	1 exam. The candidate is asked to set up a database to support a web application	Entity-Relationship Model – SQL – MySql ISTQB_CTFL_Syllabus-v4.0 ISTQB_CTFL_v4.0_Sample-Exam-A-Questions	Workshop and lecture guides	LU 3.1_Entity-Relationship Model, SQL, MySql_Lesson.pptx LU 3.1_ISTQB_CTFL_Syllabus-v4.0_Lesson_Lecture notes.PDF LU 3.1_ISTQB_CTFL_v4.0_Sample-Exam-A-Questions_v1.6_Lecture notes.PDF
3.2 HTML5, CSS3, BOOTSTRAP	5	24 hours	The didactic approach would be aimed to allows participants to understand the main applications of those sets of code using	Assignment: practical activity. The student is asked to create a web application using HTML5, CSS3 and Bootstrap.	HTML5 CSS3 Bootstrap	Workshop and lecture guides	LU 3.2_HTML5 CSS3 Bootstrap_Lesson.pptx

			practical examples, such as viewing and analyzing programming code.				
3.3 Javascript, AJAX, Typescript, GIT.	5	24 hours	The didactic approach would be aimed to allows participants to understand the main applications of those sets of code using practical examples, such as viewing and analyzing programming code.	Assignment: practical activity. The student is asked to develop a web application using frontend coding (Javascript, AJAX, Typescript) and related tools.	Javascript – Ajax	Workshop and lecture guides	LU 3.3_Javascript Ajax_Lesson.pptx
3.4 Angular, React. Jest, Mocha, Selenium	5	24 hours	The didactic approach would be aimed to allows participants to understand the main applications of those sets of	Assignment: practical activity. The student is asked to develop a web application using Angular and tools such as: React, Jest,	Angular – React – Jest – Mocha – Selenium	Workshop and lecture guides	LU 3,4_Angular – React – Jest – Mocha – Selenium_Lesson.pptx

			code using practical examples, such as viewing and analyzing programming code.	Mocha, Selenium.			
3.5 Principles of UI/UX Design. Adobe XD, Zeplng.	5	12 hours	The didactic approach would be aimed to allows participants to understand the main applications of those sets of code using practical examples, such as viewing and analyzing programming code.	Assignment: practical activity. The student is asked to build the best User Experience for the web application developed in the previous modules.	Principles of UI UX Design – Adobe XD – Zeplng	Workshop and lecture guides	LU 3.5_Principles of UI/UX Design_Lesson.pptx
3.6_Java 11, spring boot, spring data/ Hibernate. Ex. Java: Junit, Mockito	5	16 hours	The didactic approach would be aimed to allows participants to understand the main applications of	Assignment: practical activity. The student is asked to build and test a backend-side application using Java 11 and	Backend – Java 11	Workshop and lecture guides	LU 3.6_Backend_Java 11_Lesson.pptx

			those sets of code using practical examples, such as viewing and analyzing programming code.	related development tools.			
3.7_BACKEND: php, Laravel, Eloquent	5	16 hours	The didactic approach would be aimed to allows participants to understand the main applications of those sets of code using practical examples, such as viewing and analyzing programming code.	Assignment: practical activity. The student is asked to build and test a backend-side application using PHP and related development tools.	PHP – Lavarel – Eloquent	Workshop and lecture guides	LU 3.7_PHP Laravel Eloquent_Lesson.pptx
3.8_Introduction to STLC (software testing Life cycle). Unit test, end to end test.	5	8 hours	The didactic approach would be aimed to allows participants to understand the main	Assignment: practical activity. The student is asked to test an application developed in the	Introduction to STLC	Workshop and lecture guides	LU 3.8_Introduction to STLC_Lesson.pptx

	applications of STLC using practical examples, such as viewing and analyzing programming code.	previous modules.		
--	--	-------------------	--	--

1.1.7 PLO 4. Testing [e-2]⁴

4. PLO Testing [e-2]

The learner has demonstrated capability
 → to test a software application or component
 → to document test outcomes

Unit learning outcomes	Explains and distinguishes principles of software testing, common testing methods, techniques, and tools
	Writes an (automated) test on a piece of code
	Performs common test activities, applying testing and debugging techniques and tools
	Records and interprets test outcomes and writes test result documentation/ test report

1.1.7.1 Duration of Study

Recommended duration: starting from n. 2 ECTS

Often integrated with studies of PLOs: not applicable

1.1.7.2 Recommendations for Micro-credentials

N/A

1.1.7.3 Recommendations on Didactical Approach, Delivery Methods and Training Environment

Recommended didactical approach:

- Virtual Classroom

Additional comments

n/a

Recommended delivery methods:

- Lecture up to 60%
- Coding Training Lab delivered by individual/team project work up to 40%

⁴ Unit Learning outcomes are directly sourced from the ESSA Educational Profiles developed according to different EQF levels in previous WP3. For full consultation, refer to <https://softwareskills.eu/library/essa-educational-profiles-for-software-roles/>

Additional comments

It is recommended to deepen the topics presented in the Learning Units by reading publications dedicated to software testing procedures, reading websites specialized in STLC, watching online tutorials and downloading materials useful for practical exercises from reliable sources.

1.1.7.4 WBL and Follow-up Reinforcement

After learning the basic principles, terminology, and models of Software Testing, the study should focus on analysing and simulating real work-life-like tasks as, for example:

- Define and explain appropriate test methods, techniques, and tools.
- Explain and write (parts of) testing related documentation, such as a test plan, test strategy/approach, test case, test script, test scenario, test conditions.
- Setup a test environment.

1.1.7.5 Important (new) approaches and technologies to consider

N/A

1.1.7.6 Assessment

Unit learning outcome	Assessment method	Validation of prior acquired competences (skills and knowledge)
Explains and writes (parts of) testing related documentation, such as a test plan, test strategy/approach, test case, test script, test scenario, test conditions.	Assignment: practical activity. The student is asked to test a web application	N/A
Configures a test environment.	Assignment: practical activity. The student is asked to test a web application	N/A
Executes associated test cases and performs test activities related to different sorts of common tests.	Assignment: practical activity. The student is asked to test a web application	N/A
Writes test result documentation/ test report.	Assignment: practical activity. The student is asked to test a web application	

1.1.8 Learning Resources – PLO 4. Testing [e-2]

LEARNING UNIT	EQF	Duration	Didactical Approach	ASSESSMENT	TOPIC	Delivery method of the learning material	Learning materials
4.1 Fundamentals of software testing : Introduction to STLC, Unit testing, end to end testing	5	8 hours	The didactic approach would be aimed to allows participants to understand the main applications of STLC using practical examples, such as viewing and analyzing programming code.	Assignment: practical activity. The student is asked to test an application developed in the previous modules.	Introduction to STLC – Unit test – end to end test <i>ISTQB Syllabus_Lecture</i> <i>ISTQB certification sample questions</i>	Workshop and lecture guides	LU 4.1_Introduction to STLC – Unit test – end to end test_Lesson.pptx LU 4.1_ ISTQB_CTFL_Syllabus-v4.0_Lesson_Lecture notes.pdf LU 4.1_ISTQB_CTFL_v4.0_Sample-Exam-A-Questions_v1.6_Lecture notes.pdf

1.1.9 PLO 5. Documentation Production [e-2]⁵

5. PLO Documentation Production [e-2]

*The learner has demonstrated capability
→ to draft technical documentation*

Unit learning outcomes	Describes types of technical documentation
	Provides different (parts of) common technical documents, using appropriate tools (e.g., software documentation tools)

1.1.9.1 Duration of Study

Recommended duration: starting from n. 2 ECTS

Often integrated with studies of PLOs: 3 – 4 - 5

1.1.9.2 Recommendations for Micro-credentials

N/A

1.1.9.3 Recommendations on Didactical Approach, Delivery Methods and Training Environment

Recommended didactical approach:

- Virtual Classroom

Additional comments

n/a

Recommended delivery methods:

- Lecture up to 60%
- Training Lab delivered by individual/team project work up to 40%

Additional comments

It is recommended to deepen the topics presented in the Learning Units by reading publications dedicated to the various topics, reading websites specialized in software development, watching online tutorials and downloading materials useful for practical exercises from reliable sources.

⁵ Unit Learning outcomes are directly sourced from the ESSA Educational Profiles developed according to different EQF levels in previous WP3. For full consultation, refer to <https://softwareskills.eu/library/essa-educational-profiles-for-software-roles/>

1.1.9.4 WBL and Follow-up Reinforcement

After learning the basic principles, terminology, and models of Application Development, the study should focus on analysing and simulating real work-life-like tasks as, for example the student:

- Describes types of technical documentation;
- Provides different (parts of) common technical documents, using appropriate tools (e.g., software documentation tools).

1.1.9.5 Important (new) approaches and technologies to consider

N/A

1.1.9.6 Assessment

Unit learning outcome	Assessment method	Validation of prior acquired competences (skills and knowledge)
Specifies a design for an application or software (component), taking into account certain basic constraints/ requirements;	Assignment: practical activity. The student is asked to write a software documentation.	Specifies a design for an application or software (component), taking into account certain basic constraints/ requirements; Writes code and related documentation to it Applies version management
Writes documentation to related to the coding activity.	Assignment: practical activity. The student is asked to write a software documentation.	Specifies a design for an application or software (component), taking into account certain basic constraints/ requirements; Writes code and related documentation to it Applies version management

1.1.10 Learning Resources – PLO 5. Documentation Production [e-2]

LEARNING UNIT	EQF	Duration	Didactical Approach	ASSESSMENT	TOPIC	Delivery method of the learning material	Learning materials
5.1 Entity-Relationship Model, SQL, MySql	5	16 hours	The didactic approach would be aimed to allows participants to understand the main applications of those sets of code using practical examples, such as viewing and analyzing programming code.	1 exam. The candidate is asked to set up a database to support a web application	Entity-Relationship Model – SQL – MySQL	Workshop and lecture guides	LU 5.1_Entity-Relationship Model, SQL, MySql_Lesson.pptx
5.2 HTML5, CSS3, BOOTSTRAP	5	24 hours	The didactic approach would be aimed to allows participants to understand the main applications of those sets of code using practical	Assignment: practical activity. The student is asked to create a web application using HTML5, CSS3 and Bootstrap.	HTML5 CSS3 Bootstrap	Workshop and lecture guides	LU 5.2_HTML5 CSS3 Bootstrap_Lesson.pptx

			examples, such as viewing and analyzing programming code.				
5.3 Javascript, AJAX, Typescript, GIT	5	24 hours	The didactic approach would be aimed to allows participants to understand the main applications of those sets of code using practical examples, such as viewing and analyzing programming code.	Java at main	Workshop and lecture guides	LU 5.3_Java at main_Lesson.pptx	
				Javascript Ajax		LU 5.3_Javascript Ajax_Lesson.pptx	
				ISTQB_CTFL_Syllabus-v4.0		LU 5.3_ISTQB_CTFL_Syllabus-v4.0_Lesson_Lecture notes.pdf	
				ISTQB_CTFL_v4.0_Sample-Exam-A-Questions_v1.6		LU 5.3_ISTQB_CTFL_v4.0_Sample-Exam-A-Questions_v1.6_Lecture notes.pdf	

1.1.11 PLO 6. Problem management [e-2]⁶

6. PLO Problem management [e-2]

*The learner has demonstrated capability
→ to act systematically in handling incidents and problems*

Unit learning outcomes	Systematically resolves or escalates incidents and problems, resulting in a solved incident e.g., by applying techniques and tools for troubleshooting such as diagnostic tools
-------------------------------	---

1.1.11.1 Duration of Study

Recommended duration: starting from n.1,5 ECTS

Often integrated with studies of PLOs: 4

1.1.11.2 Recommendations for Micro-credentials

N/A

1.1.11.3 Recommendations on Didactical Approach, Delivery Methods and Training Environment

Recommended didactical approach:

- Virtual Classroom

Additional comments

Recommended delivery methods:

- Lecture up to 80%
- Training Lab delivered by individual/team project work up to 20%

Additional comments

It is recommended to deepen the topics presented in the Learning Units by reading publications dedicated to the various topics, reading websites specialized in coding and web development, watching online tutorials and downloading materials useful for practical exercises from reliable sources.

⁶ Unit Learning outcomes are directly sourced from the ESSA Educational Profiles developed according to different EQF levels in previous WP3. For full consultation, refer to <https://softwareskills.eu/library/essa-educational-profiles-for-software-roles/>

1.1.11.4 WBL and Follow-up Reinforcement

After learning the basic principles, terminology, and models of Application Development, the study should focus on analysing and simulating real work-life-like tasks as, for example, the student:

- Participates in a development process and solve common problems applied to the software development.

1.1.11.5 Important (new) approaches and technologies to consider

N/A

1.1.11.6 Assessment

Unit learning outcome	Assessment method	Validation of prior acquired competences (skills and knowledge)
Defines and explains appropriate test methods, techniques, and tools.	Assignment: practical activity. The student is asked to test a web application developed in the previous Learning Unit.	n/a
Explains and writes (parts of) testing related documentation, such as a test plan, test strategy/approach, test case, test script, test scenario, test conditions.	Assignment: practical activity. The student is asked to test a web application developed in the previous Learning Unit.	n/a
Configures a test environment	Assignment: practical activity. The student is asked to test a web application developed in the previous Learning Unit.	n/a
Writes test result documentation/ test report	Assignment: practical activity. The student is asked to test a web application developed in the previous Learning Unit.	n/a

1.1.12 Learning Resources – PLO 6. Problem Management [e-2]

LEARNING UNIT	EQF	Duration	Didactical Approach	ASSESSMENT	TOPIC	Delivery method of the learning material	Learning materials
6.1 Fundamentals of software testing: Introduction to STLC, Unit testing, end to end testing.	5	8 hours	By alternating between theory and practical activities, students will be able to understand the fundamentals of software testing, with reference to software testing life cycle and to Unit Testing, software carried out during the development of an application.	Assignment: practical activity. The student is asked to test a web application developed in the previous modules.	Introduction to STLC	Workshop and lecture guides	LU 6.1_Introduction to STLC_Lessonpptx

1.1.13 PLO 7. Professional related competences [EQF5]⁷

7. PLO Profession related competences [EQF5]

*The learner has demonstrated capability
→ to apply profession related skills*

Unit learning outcomes	Masters common ICT knowledge
	Explains the principles, related concepts, advantages and disadvantages of a new technology. Applies and reports on basic methods, techniques and tools related to a new technology.
	Applies and reports on measures, methods, tools and techniques related to security
	Applies and reports on measures, methods, tools and techniques related to software lifecycle processes
	Is aware of basic ethical considerations and issues

1.1.13.1 Duration of Study

Recommended duration: starting from n.1 ECTS

Often integrated with studies of PLOs: 1 – 8 - 9

1.1.13.2 Recommendations for Micro-credentials

This PLO should be an integral part of the initial studies for students with no prior knowledge of professional-related competences useful to work in complex organizations embedded in innovative markets.

1.1.13.3 Recommendations on Didactical Approach, Delivery Methods and Training Environment

Recommended didactical approach:

- Presence Classroom
- Virtual Classroom
- Blended
- e-learning
- Work placement

Additional comments

⁷ Unit Learning outcomes are directly sourced from the ESSA Educational Profiles developed according to different EQF levels in previous WP3. For full consultation, refer to <https://softwareskills.eu/library/essa-educational-profiles-for-software-roles/>

n/a

Recommended delivery methods:

- Lecture up to 100%

Additional comments

It is recommended to deepen the topics presented in the Learning Units by reading publications dedicated to the various topics, reading websites specialized in innovation, project management and team collaboration.

1.1.13.4 WBL and Follow-up Reinforcement

N/A

1.1.13.5 Important (new) approaches and technologies to consider

- Business Process and Business Architecture understanding and mapping tools
- Product/service design and innovation management

1.1.13.6 Assessment

Unit learning outcome	Assessment method	Validation of prior acquired competences (skills and knowledge)
Masters common ICT knowledge	1 test (multiple choice questions) on the characteristics of the main IT technologies.	n/a
Works in project settings, applies project management methods and tools	Assignment: practical activity. The student is invited to apply the Agile methodology in the development of a web application.	n/a
[Security skills] Applies and reports on methods, tools and techniques related to security	Assignment: practical activity. The student is asked to test the security of a web application	n/a
[Software life cycle skills] Applies and reports on methods, tools and techniques related to software lifecycle processes	Assignment: practical activity. The student is asked to test a web application developed in the previous modules.	n/a
[Ethical awareness skills] Is aware of basic ethical considerations and issues	1 test (multiple choice questions) on the characteristics of the main IT technologies.	n/a

1.1.14 Learning Resources – PLO 7. Profession related competence [EQF5]

LEARNING UNIT	EQF	Duration	Didactical Approach	ASSESSMENT	TOPIC	Delivery method of the learning material	Learning materials
7.1 Introduction to ICT and Digital Transformation tech enablers	5	4 hours	The didactic approach would be aimed to allows participants to understand ICT culture using practical example and storytelling of case histories.	1 test (multiple choice questions) on the characteristics of the main IT technologies.	Introduction to ICT and Digital Transformation tech enablers	Workshop and lecture guides	LU 7.1_Introduction to ICT and Digital Transformation tech enablers_Lesson.pptx
7.2 Agile Project Management + Scrum + collaboration tools	5	8 hours	The didactic approach would be aimed to allows participants to understand the Agile and SCRUM culture and framework using practical examples and exercises.	Assignment: practical activity. The student is invited to apply the Agile methodology in the development of a web application.	Agile PM and SCRUM	Workshop and lecture guides	LU 7.2_Agile PM and SCRUM_Lesson.pptx
7.3 Fundamentals of cybersecurity	5	8 hours		Assignment: practical activity. The student is required to take a computer	Fundamentals of Cybersecurity	Workshop and lecture guides	LU 7.3_Fundamentals of Cybersecurity_Lesson.pptx

				security test on a series of web applications.			
7.4 Introduction to STLC (software testing Life cycle). Unit test, end to end test.	5	8 hours	The didactic approach would be aimed to allows participants to understand the main applications of STLC using practical examples, such as viewing and analyzing programming code.	Assignment: practical activity. The student is asked to test a web application developed in the previous modules.		Workshop and lecture guides	LU 7.4_Introduction to STLC_Lesson.pptx

1.1.15 PLO 8. Soft competences [EQF5]⁸

8. PLO Soft competences [EQF5]

*The learner has demonstrated capability
→ to apply soft skills*

Unit learning outcomes	Works together with others in a team
	Communicates with peers, colleagues, supervisors and/or relevant others, appropriately to the context, using conventions that are relevant to professional practice. Explains and gives instruction.
	Masters the English language at level B2. Can understand the main ideas of complex text on both concrete and abstract topics, including technical discussions in his/her field of specialisation
	Distinguishes and analyses fairly complex and unpredictable problems. Solves these problems systematically and in a creative way, using existing procedures and guidelines and own solutions by identifying and using data.
	Exercises self-management within the guidelines of contexts that are usually predictable, but are subject to change. Is able to cope with limited change and to adapt to a certain level of variety in the workplace. Copes with pressure and stress setbacks and maintains composure. Shows some initiative and carries responsibility for the results of own activities, work and or study. Works correctly and carefully.
	Realises learning and personal development on request, where necessary with support, through self-reflection and external- and self-evaluation of own (learning) results.

1.1.15.1 Duration of Study

Recommended duration: starting from n.0,5 ECTS

Often integrated with studies of PLOs: 9

1.1.15.2 Recommendations for Micro-credentials

This PLO should be an integral part of the initial studies for students with no prior knowledge of professional-related competences useful to work in complex organizations embedded in innovative business.

1.1.15.3 Recommendations on Didactical Approach, Delivery Methods and Training Environment

Recommended didactical approach:

⁸ Unit Learning outcomes are directly sourced from the ESSA Educational Profiles developed according to different EQF levels in previous WP3. For full consultation, refer to <https://softwareskills.eu/library/essa-educational-profiles-for-software-roles/>

- Presence Classroom
- Virtual Classroom
- Blended
- e-learning
- Work placement

Additional comments

n/a

Recommended delivery methods:

- Lecture up to 100%

Additional comments

It is recommended to deepen the topics presented in the Learning Units by reading publications dedicated to the various topics, reading websites specialized in programming, watching online tutorials and downloading materials useful for practical exercises from reliable sources.

1.1.15.4 WBL and Follow-up Reinforcement

N/A

1.1.15.5 Important (new) approaches and technologies to consider

N/A

1.1.15.6 Assessment

Unit learning outcome	Assessment method	Validation of prior acquired competences (skills and knowledge)
Teamwork skills] Manages teamwork processes and facilitates collaboration to reach common objectives, e.g., handles conflicts, negotiates, motivates, and persuades.	The student is called to work on a development project, creating a work group and collaborating with other people through the use of enterprise social collaboration tools.	n/a
Communication skills] Communicates with peers, colleagues, supervisors and or relevant other, specialists and non-specialists, and clients, appropriately to the scientific and professional community, using conventions which are relevant. Applies communication to the objective and the target group.	The student is called to work on a development project, creating a work group and collaborating with other people through the use of enterprise social collaboration tools. The student is called to work on a development project, creating a work group and collaborating with other people through the use of enterprise social collaboration tools.	n/a
Problem solving skills] Identifies and analyses complex and unpredictable	The student is called to work on a development project, creating a work group and collaborating with	n/a

<p>problems Solves these problems in a tactical, strategic and creative way by selecting and using data and by using one's creativity, flexibility and inventiveness.</p>	<p>other people through the use of enterprise social collaboration tools.</p>	
<p>[Self-management skills] Realises personal development on one's own initiative, by reflecting on and evaluating personal (learning) results.</p>	<p>The student is called to work on a development project, creating a work group and collaborating with other people through the use of enterprise social collaboration tools.</p>	<p>n/a</p>

1.1.16 Learning Resources – PLO 8. Soft competences [EQF5]

LEARNING UNIT	EQF	Duration	Didactical Approach	ASSESSMENT	TOPIC	Delivery method of the learning material	Learning materials
8.1 Team Collaboration (soft Skill)	5	8 hours	The didactic approach would be aimed to allows participants to understand the main team collaboration principles and practices, using practical examples, case study analysis and exercises.	Assignment: practical activity. The student is called to work on a development project, creating a work group and collaborating with other people through the use of enterprise social collaboration tools.	Team Collaboration and file versioning	<ul style="list-style-type: none"> • Formative quizzes • Workshop and lecture guides 	LU 8.1_Team Collaboration and file versioning_Lesson.pptx
8.2 Introduction to STLC (software testing Life cycle)	5	8 hours	The didactic approach would be aimed to allows participants to understand the main applications of STLC using practical examples, such as viewing and analyzing	Assignment: practical activity. The student is asked to test a web application developed in the previous modules.	Introduction to STLC	<ul style="list-style-type: none"> • Formative quizzes • Workshop and lecture guides 	LU 8.2_Introduction to STLC_Lesson.pptx

| | | programming code. | | |

1.1.17 PLO 9. Functioning in organisations [EQF5]⁹

9. PLO Functioning in organisations [EQF5]

*The learner has demonstrated capability
→ to function in an organisational context*

Unit learning outcomes	Explains the basics of organisation theory and behaviour
	Describes the relationship between business and IT
	Works in an organisational context under specific direction with limited autonomy and responsibility e.g., at the level of a trainee, junior or assistant
	Works in project settings, applies project management methods and tools
	Writes a report on functioning in the organisation

1.1.17.1 Duration of Study

Recommended duration: starting from n.0,5 ECTS

Often integrated with studies of PLOs: 8

1.1.17.2 Recommendations for Micro-credentials

This PLO should be an integral part of the initial studies for students with no prior knowledge of teamwork collaboration.

1.1.17.3 Recommendations on Didactical Approach, Delivery Methods and Training Environment

Recommended didactical approach:

- Virtual Classroom

Additional comments

n/a

Recommended delivery methods:

- Lecture up to 100%

Additional comments

⁹ Unit Learning outcomes are directly sourced from the ESSA Educational Profiles developed according to different EQF levels in previous WP3. For full consultation, refer to <https://softwareskills.eu/library/essa-educational-profiles-for-software-roles/>

It is recommended to deepen the topics presented in the Learning Units by reading publications dedicated to the various topics, reading websites specialized in teamwork collaboration and communication & collaboration tools and platforms.

1.1.17.4 WBL and Follow-up Reinforcement

N/A

1.1.17.5 Important (new) approaches and technologies to consider

N/A

1.1.17.6 Assessment

Unit learning outcome	Assessment method	Validation of prior acquired competences (skills and knowledge)
Explains basics of organisation theory and behaviour	Assignment: practical activity. The student is called to work on a development project, creating a work group and collaborating with other people through the use of enterprise social collaboration tools.	n/a
Works in an organisational context under specific direction with limited autonomy and responsibility	Assignment: practical activity. The student is called to work on a development project, creating a work group and collaborating with other people through the use of enterprise social collaboration tools.	n/a

1.1.18 Learning Resources – 9. PLO Functioning in organisation [EQF5]

LEARNING UNIT	EQF	Duration	Didactical Approach	ASSESSMENT	TOPIC	Delivery method of the learning material	Learning materials
9.1 Team Collaboration	5	8 hours	The didactic approach would be aimed to allows participants to understand the main team collaboration principles and practices, using practical examples, case study analysis and exercises.	Assignment: practical activity. The student is called to work on a development project, creating a work group and collaborating with other people through the use of enterprise social collaboration tools.	Team collaboration and file versioning	Workshop and lecture guides	LU 9.1_Team Collaboration and file versioning_Lesson.pptx
9.2 ICT and Digital Transformation tech enablers	5	8 hours	The didactic approach would be aimed to allows participants to understand ICT culture using practical example and storytelling of case histories.	1 test (multiple choice questions) on the characteristics of the main IT technologies.	Introduction to ICT and Digital Transformation Tech Enablers	Workshop and lecture guides	LU 9.2_Introduction to ICT and Digital Transformation tech enablers_Lesson.pptx

www.softwareskills.eu



Co-funded by the
Erasmus+ Programme
of the European Union

The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.