U.Porto Journal of Engineering, 9:3 (2023) 209-222
ISSN 2183-6493
DOI: 10.24840/2183-6493_009-003_001898

Received: 29 October, 2022
Accepted: 28 December, 2022
Published: 28 April, 2023

An innovative tailored instructional design for computer programming courses in engineering

Dominique Persano Adorno[1], Tuba Ugras[2], James Sunney Quaicoe[3], Veselina Jecheva[4], Abiodun A. Ogunyemi[5], Merja Bauters[6], Angel Toshkov[7], Yasin Ortakci[8], Kasim Ozacar[9], Ferhat Atasoy[10], Daniele Peri[11], Slavko Kocijancic[12], David Rihtaršič[13], Špela Cerar[14], Huseyin Uvet[15]

[1]Department of Physics and Chemistry, University of Palermo, Palermo, Italy (dominique.persanoadorno@unipa.it) ORCID 0000-0001-7655-1114

[2]Faculty of Education, Yildiz Technical University, Istanbul, Turkey, (tugras@yildiz.edu.tr) ORCID 0000-0001-8241-5696

[3]School of Digital Technologies, Tallinn University, Tallinn, Estonia, (paasanni@tlu.ee) ORCID 0000-0003-0287-6594

[4]Faculty of Computer and Technical Sciences, Burgas Free University, Burgas, Bulgaria, (vessi@bfu.bg) ORCID 0000-0002-3431-9269

[5]School of Digital Technologies, Tallinn University, Tallinn, Estonia, (abnogn@tlu.ee) ORCID 0000-0002-1851-0442

[6]School of Digital Technologies Tallinn University, Tallinn, Estonia, (bauters@tlu.ee) ORCID 0000-0001-8501-5751

[7]Faculty of Computer and Technical Sciences, Burgas Free University, Burgas, Bulgaria, (angel_toshkov@abv.bg) ORCID 0000-0003-3798-6283

[8]Faculty of Engineering, University of Karabuk, Karabuk, Turkey, (yasinortakci@karabuk.edu.tr) ORCID 0000-0002-0683-2049

[9]Faculty of Engineering, University of Karabuk, Karabuk, Turkey, (kasimozacar@karabuk.edu.tr) ORCID 0000-0001-7637-0620

[10]Faculty of Engineering, University of Karabuk, Karabuk, Turkey, (ferhatatasoy@karabuk.edu.tr) ORCID 0000-0002-1672-0593

[11]Department of Engineering, University of Palermo, Palermo, Italy (daniele.peri@unipa.it) ORCID 0000-0002-8763-7199

[12]Faculty of Education, University of Ljubljana, Ljubljana, Slovenia, (slavko.kocijancic@pef.uni-lj.si) ORCID 0000-0003-4597-9729

[13]Faculty of Education, University of Ljubljana, Ljubljana, Slovenia, (david.rihtarsic@pef.uni-lj.si) ORCID 0000-0003-3877-1137

[14]Faculty of Education, University of Ljubljana, Ljubljana, Slovenia, (Spela.Cerar@pef.uni-lj.si) ORCID 0000-0003-0229-9539

[15]Faculty of Mechanical Engineering, Yildiz Technical University, Istanbul, Turkey, (huvet@yildiz.edu.tr) ORCID 0000-0003-0392-982X

Abstract

Industry 4.0 and 5.0 topics are emerging fields and have seen rising demand recently. There is a critical need, on the other hand, for improved methods of instructing programming languages since a growing lack of student motivation during the pandemic has had a deleterious influence on the education of programmers. In this context, online/hybrid computer programming courses must be addressed with innovative solutions to support the field with well-educated professionals. In this paper, we present a case study to propose an innovative tailored instructional design for the online/hybrid learning environments for programming courses in engineering faculties. To develop the instructional design, the Kemp Instructional Design Model was followed. The instructional design is a result of the main outputs of the RECOM “Redesigning Introductory Computer Programming Using Innovative Online Modules” project, which aims to bridge the gap between the existing course design in programming courses and the needs of "Covid” and “post-Covid” generation students.

Author Keywords. Hybrid Learning, Course Design, Undergraduate Innovative Education, Programming Languages

Type: Research Article
Open Access Peer Reviewed

1. Introduction

The Covid-19 Pandemic forced the education system to change radically. Almost all universities worldwide were suddenly forced to shift all their education to virtual learning in response to the pandemic. Therefore, support for lecturers' professional development has become crucial because course design in online/hybrid teaching is more critical than in traditional education settings. Online/hybrid teaching environments bring major challenges to faculty members; the most significant is the complexity involved in moving a face-to-face course into the online/hybrid platforms (**Hampsten 2021; Rapanta et al. 2020; Singh, Steele & Singh 2021**). Since the traditional education system mostly prevailed in many European countries, teachers and students in online education encountered several problems, especially in engineering departments, due to the dense curriculum on hand-zone and practical lessons. Other problems centred around the experience in online teaching/learning, students' collaboration, a lack of support from lecturers, the complexity of materials provided, and an inappropriate homework environment (**Trammell & LaForge 2017; Akcil & Bastas 2020; Hehir et al. 2021; Hu, Sang & Meng 2021; Nortvig, Petersen & Balle 2018**). These facts reduced satisfaction and enthusiasm for both students and educators.

Active learning is considered the natural framework to develop occasions of learning in terms of the active construction of meaningful knowledge (**Rossi et al. 2021; Bybee 2010**). Some instructional strategies, such as gamification, inquiry, project-based activities, and flipping the classroom, have an important role in supporting active learning (**Saleem, Noori & Ozdamli 2022; Jaftha, Zahra-Micallef & Chircop 2021; Dichev & Dicheva 2017; Vargas-Macías, Rodríguez-Hernandez & Sánchez-Sáenz 2020; Rincon-Flores & Santos-Guevara 2021; Persano Adorno 2022; Davies, Dean & Ball 2013; Strayer 2012**). In particular, a student-based learning approach promotes greater engagement of learners. In hybrid learning environments, online educational materials, digital learning tools, and opportunities for interaction are combined with more traditional classroom methods (**Förster et al. 2021**).

The “Redesigning Introductory Computer Programming Using Innovative Online Modules” (RECOM) project, funded by the European Union within the Erasmus+ KA226 program (Partnership for Digital Education Readiness), aims to bridge the gap between the existing national curricula related to engineering studies and the needs of “Covid” and “post-Covid” generation students. It involves six universities – Yildiz Technical University, Burgas Free University, University of Palermo, University of Ljubljana, University of Karabuk, Tallinn University and the Board of European Students of Technology (BEST) organisation. The project focused on the Computer Programming and Algorithms topic, which is a core element for all engineering disciplines and is a complicated course both to teach and learn. In the project, appropriate online/hybrid teaching methods for this lecture will be designed and integrated with online technologies to improve student's learning motivation and interest, maintain students' concentration, and enhance students' learning.

The main outputs of the RECOM project include (i) a course design framework for online/hybrid computer programming courses, (ii) a web-platform with mobile access for the course content, (iii) educational materials in the form of serious games, interactive video tutorials, e-quizzes and case studies. It is still an ongoing project, however, the partner consortium has already completed the course design, the web-platform, the educational materials and currently working on the evaluation process.

The RECOM project seeks to create a systematic student-centric approach to online/hybrid instruction for courses on programming in engineering for Bachelor level students. In line with this objective, this paper sets out to explore how the project deliverables - course design, web-platform and educational materials harmonise to evolve the student-centric learning approach.

The educator's skills, training, and commitment are critical to implementing an effective online/hybrid pedagogy. Thus, an applied engineering course, such as a programming course, requires different strategies for educators to present content, interact with students, and assess course outcomes (**Förster et al. 2021; Omeh, Olelewe & Nwangwu 2022**). A solid understanding of the learning process is essential responding to the enormous global changes in higher education. The answers to better teaching and learning strategies require international collaborations from various disciplines and backgrounds that focus on the same problem. Within the wide partnership created by the project we shared and discussed experiences. In that sense, this paper is intended to contribute to the online/hybrid learning environments for programming language courses in engineering faculties.

2. Methodology and Theoretical Framework

In this study, the research design is a case study that is structured within the framework of the Kemp Instructional Design model. The Kemp Instructional Design Model is an innovative approach to instructional design since it has a non-linear structure with interconnected components (**Morrison et al. 2010**). Thus, it has a circular structure that is realised through nine linked interdependent components see Figure 1. This allows great flexibility in the design of training plans, as with this model it is possible to start the design process from each of the nine components or stages, without having to adhere to the constraints required by following a linear approach. Depending on the learning system design process, several stages may be started simultaneously, and some design stages may be dropped. Because of the interrelationship between the components, the design process becomes cyclical.

In this approach, training designers will need to consider not only training objectives but also a number of other factors, including the needs and desires of learners, training content and activities (including tasks and procedures), training resources and support services, as well as assessment and evaluation tools and methods.

The cyclical approach proposed in Kemp's model directs designers to use the learner's perspective so that learners' goals, priorities, and limitations are taken into account. Kemp Instructional Design's nine key components, which are designed to focus on learners throughout the design process, are much more detailed and nuanced than those offered in other familiar systems.

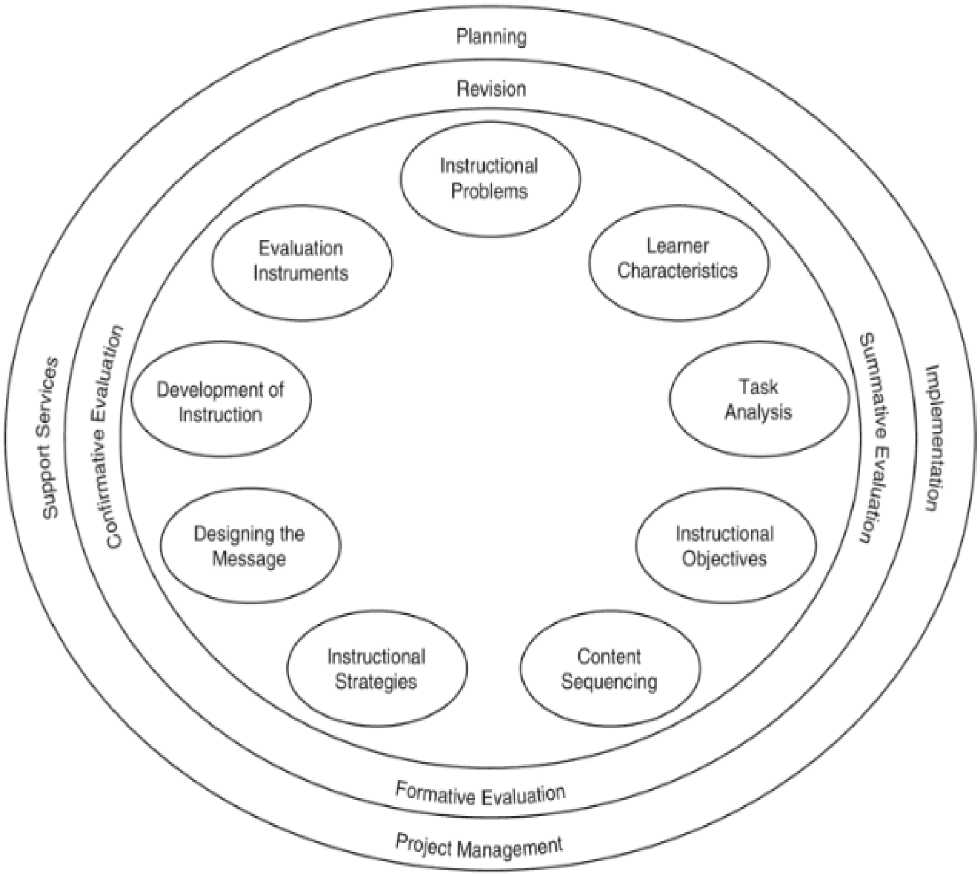

Figure 1: The Kemp Instructional Design Model (**Morrison et al. 2010**)

The Kemp ID model has nine components within the inner cycle, namely, (i) Instructional problem, (ii) Learner characteristics, (iii) Task analysis, (iv) Instructional objectives, (v) Content sequencing, (vi) Instructional strategies, (vii) Designing the message, (xiii) Development of instruction, and (ix) Evaluation instruments (**Morrison et al. 2010**). Regarding the instructional problem component, we focused on "how to design effective programming languages courses in online/hybrid environments for engineering students". For the learner characteristics component, we run a questionnaire for both students and teachers to grasp the needs and desires of learners as well as the features of the desired learning environment. We examined the existing course design plans as a data collection method within the scope of the task analysis component. They allowed us to scaffold the course design -the first iteration- in terms of the components of instructional objectives, content sequencing, instructional strategies, designing the message, development of instruction, and development of evaluation instruments. As a result of the first iteration, we proposed a course design template, which is presented in detail in a conference paper (**Uvet et al. 2022**). In the second iteration, we continued to work on the same components through the course design template. In other words, we filled out the template by developing the required information in terms of each related component. As a result of the second iteration, we created an instructional design that is being presented in this study. As the next step, we will evaluate it -as the outer cycle of the Kemp ID model- via expert evaluations and a pilot training with students.

3. Results and Discussion

As the results of the study, the instructional design outline and resources are presented below. The instructional design includes a detailed course plan, a web-platform, and related educational materials.

3.1. The Course Plan

For developing the course plan, we utilized the course design template, which is the result of our preliminary study (**Uvet et al. 2022**) in which we implemented the first iteration of the Kemp ID model. The preliminary study started by using a survey to elicit information both from Bachelor level engineering students and teachers. In all, 372 students were sampled from 19 European countries, and 47 teachers from five European countries participated in the data collection. Results showcased the remarkable benefit of online programming language classes was the variety of learning elements, for both teachers (47%) and students (33%). Students preferred video tutorials (58%), simulations (52%), and external resources (49%) as the educational elements that support lessons. Furthermore, teachers also stated they would mostly employ project-based learning (49%), gamification (45%), simulation-based learning (34%), and flipped classroom (30%) activities in their classes. The noteworthy challenge was providing insufficient feedback, as stated by teachers (77%) and students (46%). The preliminary study also examined the existing course plans in terms of content, learning objectives, teaching methods, and evaluation methods.

Therefore, as the next step, the specifications for the course design and the course plan template were formulated. Figure 2 provides the schema defining the specifications. From the schema, three key sections of the course plan are shown: Pre-Instructional activities, Instructional sessions, and Post-Instructional activities. Pre-Instructional activities are proposed so that teachers embed themselves in the planning of the lesson with the student in mind. The pedagogical requirements are the following: (i) Topic and/or subtopics, (ii) Specific objectives, (iii) the environment for the learning, and (iv) resources/material for the lesson. During the instructional sessions, the teacher is expected to align the pre-instructional projections to the instructional and post-instructional delivery sessions before the lesson delivery. The emphasis here is on aligning lesson planning, lesson delivery, and lesson assessment to be centred around learners' support, with the teacher-centred roles relegated to the background as a facilitator and a mentor. Therefore, in planning and delivering the lesson, considerations include the following: (i) The lesson delivery, (ii) Teacher-centred activities, (iii) Formative assessment, (iv) Motivation techniques, and (v) Lesson conclusion. Post-Instructional activities are considered essential for teachers to showcase empathy and promote individual learning support. Thus, the framework proposes the following post-instructional activities: (i) Summative assessment, (ii) Remedial plan for students, and (iii) Preparing students for the next lesson.

Based on the schema, a template for the course design was designed. This addressed all the specifications defined in the schema (see Figure 3). Consequently, we filled this template to construct the course plan for programming languages courses in engineering faculties, delivered in online/hybrid environments. Figure 3 shows the course plan with the first two topics. The topic list is given in Table 1.

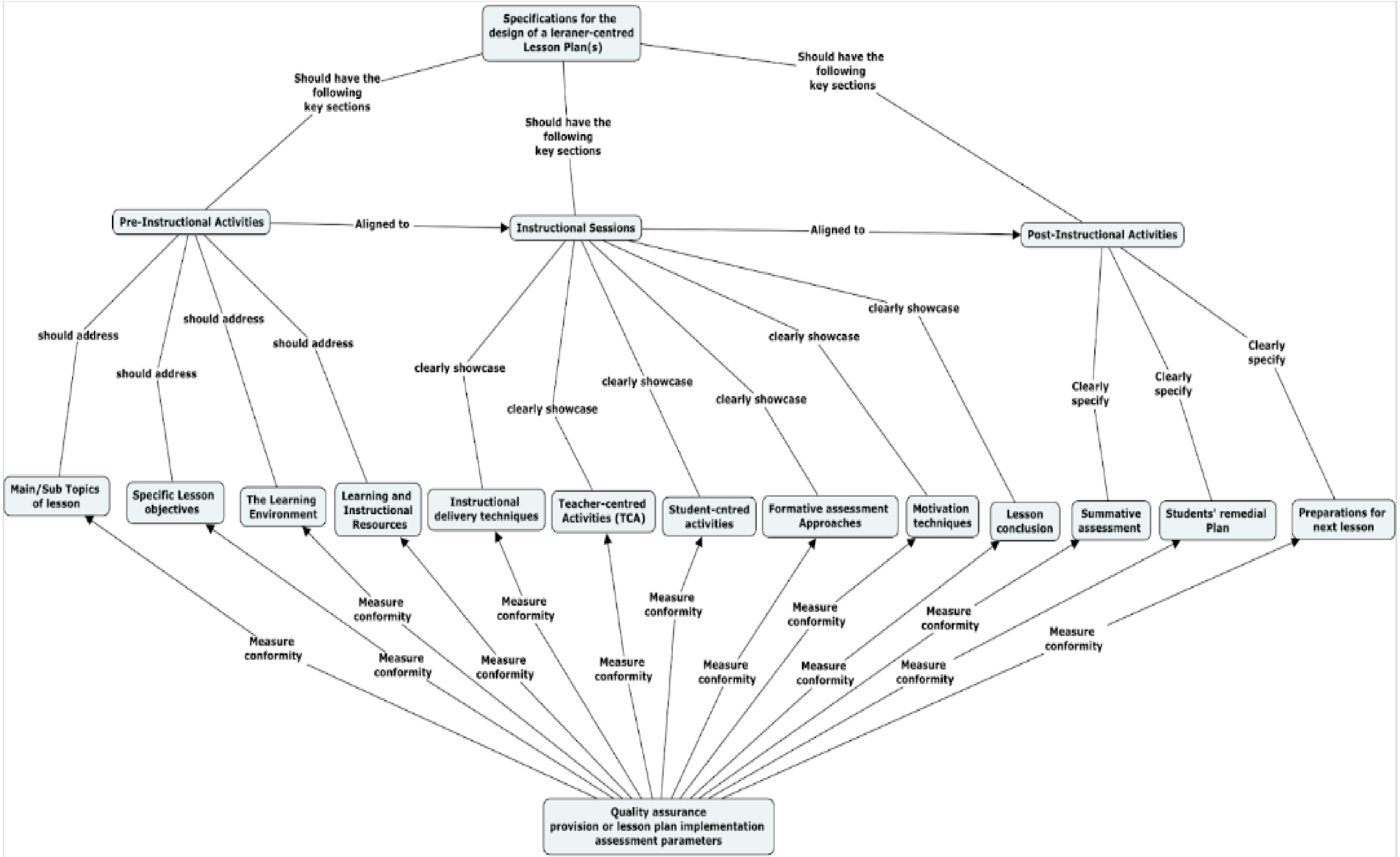

Figure 2: Schema of the requirements of the design of a learner-centred course design.

Topic Number	Topic Name
1	Overview of Computer Programming
2	Variable Concepts and Operators
3	Decision Making I
4	Decision Making II
5	Repetition and Loop statements
6	Modular Programming I
7	Modular Programming II
8	Array Data Collection
9	Pointers and References
10	Strings

Table 1: Topics in computer programming course

In sum, the course plan has features, especially in terms of instructional strategy issues that are closely related to student-based learning approaches such as feedback, gamification, and flipping the classroom. In this course plan, conscious efforts are made to put the teacher in the role of facilitator and promote active student learning.

Teaching traditional programming languages like C/C++ is not a trivial task. It requires algorithmic thinking, mathematics, problem-solving and real-life knowledge and skills, which students do not easily obtain. We also analyzed the existing course plans from five European countries. According to the course plan analysis, we identified the weekly topics, learning objectives, and related pedagogical approaches. The applied methodology includes instructional components that analyze engineering problems and measure or simulate students' performance. In addition, the selected approach aims at developing flexible instructional design for content choice, delivery, and assessment within the framework of the preliminarily defined learning objectives.

With reference to the theoretical underpinnings of the project, the Kemp ID Model, a learner-centred course plan was proposed for use by lecturers/instructors to adapt for designing course plan(s). It consists of learning objectives, followed by designing the course content in terms of practice skills, teaching methods, and assessment. This step is very important since online/hybrid teaching is different from face-to-face in terms of difficulty in engaging students' attention.

Pre-Instructional Activities					Instructional Session: pedagogical considerations						Post-Lesson activities		
Main Topic	Sub-topic	Specific Objectives/Expected learning outcomes	Lesson Environment and/Mode	Assigned resources and materials	Instructional delivery (Activities outlines)	Teacher-centred Activities (TCA)	Student-centred activities(SCA)	Formative assessment approaches (e.g feed and other forms of realtime instructional support for students	Motivation Techniques (rewards, support and encouragments etc)	Lesson Conclusion	Summative assessment and feedback activitiies	Students' Remedial Plan	Preperation for next lessoon (e.g flipping the class)
1. Overview of Computer Programming	Computer software basics	Describe user instructions to enable hardware devices and software program	Online/Hybrid	Online video resources on Youtube (Link should be added)	Open-ended questions that are connected to learning outcomes	Directed Discussion	free writes, one-sentence summary,	(1) Questions/Answers on IDE environments / Computer Systems (2) Feedback on the first program creation (Hello Word) (3) Comments from students approx 10 min. (4) LMS for streaming comments	(1) Student oral feedback through online lesson environment and rewarding the first 3 correct answers (2) Motivating the students for asking questions and rewarding the best 3 students questions	(1) Summarize the current lecture content (2) Announce the assignment for the lecture (3) Brief introduction to the next lecture (Relevant chapter from the book should be shared)	(1) Homework or (2) Take home exam (3) Supplying feedback form for the lecture which has basic indicators such as difficulty of the lecture, how well the lecture is explained, how well they learn the subject, etc.	(1) Revisiting the recorded video of lecture (2) Office hours for the Lecturer or the teaching assistant (3) Sharing the lecture notes (4) Counseling (5) Lesson tutorials/drills/simulations (Links should be shared) (6) Off-topic meetings (1 hour meeting with students to discuss about anything except the lecture itself)	(1) Videos of previous year's lecture (Link should be shared) (2) Reference book chapter (3) Homework assignment which includes the basics of the next lecture or (4) Basic surprise quiz in the beginning of the next lecture (the contents should be generic)
	Program Development Environments	Use development environments and their programming tools to create the program or software product		Online video resources on Youtube (Link should be added)	Establishing programming environment	Direct Instruction	Student will install IDE on their own computers						
	Compilation and execution	Use development environments and their programming tools to create the program or software product		Online video resources on Youtube (Link should be added)	Running programming environment.	Direct Instruction	Students do the tasks,						
	Algorithm basic concepts	Apply techniques of building blocks to develop basic programming steps in the derivation of algorithms		Online video resources on Previously recorded videos (Link should be added)	Classify concepts, define vocabulary, scaffold steps	Directed Discussion	Students do the task and team/pairs evaluate each other outcome						
	Block diagrams	Apply techniques of building blocks to develop basic programming steps in the derivation of algorithms		Online video resources on Previously recorded videos (Link should be added)	Showing and explaining examples - Example (1)	Interactive Lecture with Video-Tutorial (1)	free writes, one-sentence summary, one minute papers						
	Structure of C++ language	Explain various programming elements and sections of C++ language		Online video resources on Previously recorded videos (Link should be added)	Open-ended questions that are connected to learning outcomes	Direct Instruction	Open-ended questions that are connected to learning outcomes						
	Testing and driving a computer program in Console	Test and run programs using fundamental user interface of C++ language		LMS preparation	Group creation	List and group the students	Student practice on console input and output						
2. Variable Concepts and Operators	Basic data types	Learn basic data types and memory concept in C++ programming	Online/Hybrid	Textbook / Manuals / Mooc (Relevant chapters should be announced)	Open-ended questions that are connected to learning outcomes	Directed Discussion	Students focus on their learning process through application with Mobile Game (1)	Group work/team project with Case-Study (1) to make students applying course knowledge to produce something	Reward System for assignments (League of programming concept - LOP)	(1) Summarize the current lecture content (2) Announce the assignment for the lecture (3) Brief introduction to the next lecture (Relevant chapter from the book should be shared)	(1) Homework or (2) Take home exam (3) Supplying feedback form for the lecture which has basic indicators such as difficulty of the lecture, how well the lecture is explained, how well they learn the subject, etc.	(1) Revisiting the recorded video of lecture (2) Office hours for the Lecturer or the teaching assistant (3) Sharing the lecture notes (4) Counseling (5) Lesson tutorials/drills/simulations (Links should be shared) (6) Off-topic meetings (1 hour meeting with students to discuss about anything except the lecture itself)	(1) Videos of previous year's lecture (Link should be shared) (2) Reference book chapter (3) Homework assignment which includes the basics of the next lecture or (4) Basic surprise quiz in the beginning of the next lecture (the contents should be generic)
	Constants and variables	Demonstrate and apply techniques of declaring and initializing variables		Textbook / Manuals / Mooc (Relevant chapters should be announced)	Open-ended questions that are connected to learning outcomes	Directed Instruction	Student practice on computer						
	Numeric types, operations	Identify all numeric types and operations.		Textbook / Manuals / Mooc (Relevant chapters should be announced)	Multiple-choice items, solving a problem, comparing and filling in lecture notes,	presenting material and answering student questions	Student practice on computer						
	Assignments	Learn assignment operators' syntax to assign values to variables		Textbook / Manuals / Mooc (Relevant chapters should be announced)	Open-ended questions that are connected to learning outcomes	Demonstration w/IDE Student	Suprise questions (Equiz-1) to motivate students to do the readings						
	Type casting	Choose appropriate primitive data types and become familiar with the casting technique to enable the conversion of one data type into another		Textbook / Manuals / Mooc (Relevant chapters should be announced)	Applying course content to real-world engineering examples	Demonstration w/IDE	Student practice on computer						

Figure 3: A sample from the course plan

The Web-platform

The RECOM web-platform with mobile access has been developed using the Moodle infrastructure. Thanks to its integrated set of tools that are designed to support a variety of kinds of online interactions between students and teachers, Moodle has been adopted by many educators and institutions around the world (**Kumar, Gankotiya & Dutta 2011**). Moodle is an open-source, robust, secure, and integrated learning platform for educators, students, and administrators to create unique learning environments (**Moodle 2022**). It is widely used on more than 169 thousand websites worldwide, includes more than 43 million courses in 241 countries, has more than 351 million users, and continues to be updated daily (**Moodle Statistics 2022**). Thanks to being open source, the development of necessary add-ons and the creation of customised solutions on Moodle also contribute to the growth of users. Additionally, Moodle is compatible with practically all platforms and browsers since it is designed to be mobile-friendly and is built on the web (**Aberdour 2013**). Taking into account the advantages outlined above, we decided on Moodle as the project's learning platform.

The RECOM web-platform developed using Moodle aims to provide student-centred learning activities. Instead of attending a course once a week, it enables students to take part in a variety of engaging activities throughout the week. Five interactive video tutorials, five mobile games, five e-quizzes, and five case studies are the main educational resources used to provide the activities.

The main interface of the course content on the web-platform is shown in Figure 4. Both teachers and students use the same user interface and reach the resources by clicking the links on the course page. Additionally, teachers have an editing mode through which they are able to edit, add, or delete content and organize online exam or quiz activities, etc. (See Figure 5)

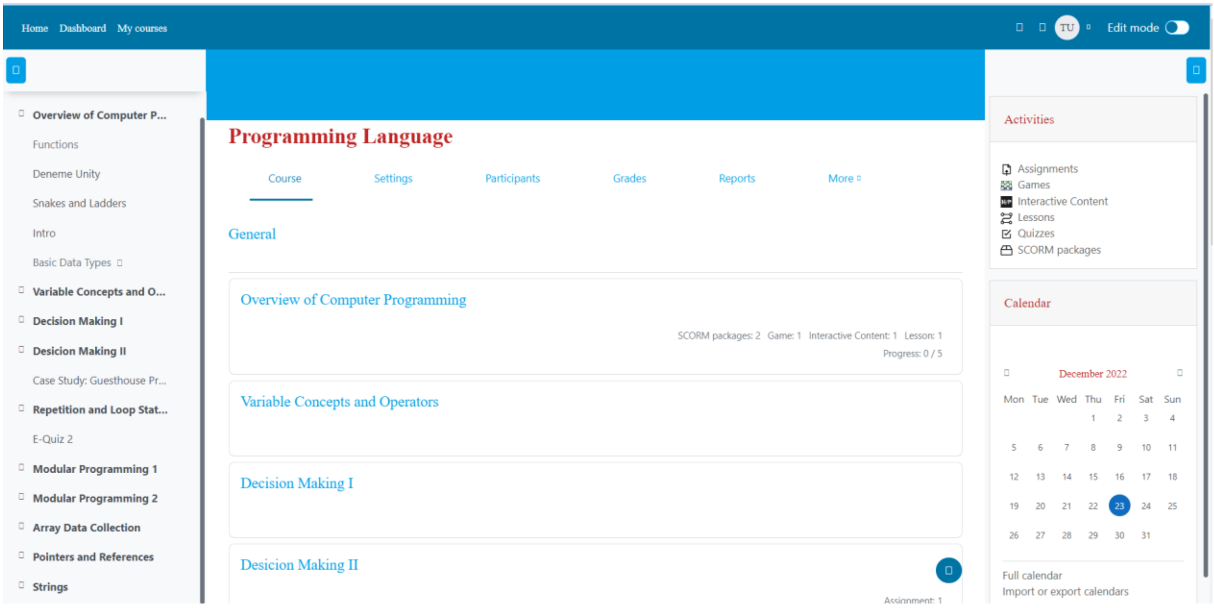

Figure 4: The interface of the course content on web-platform

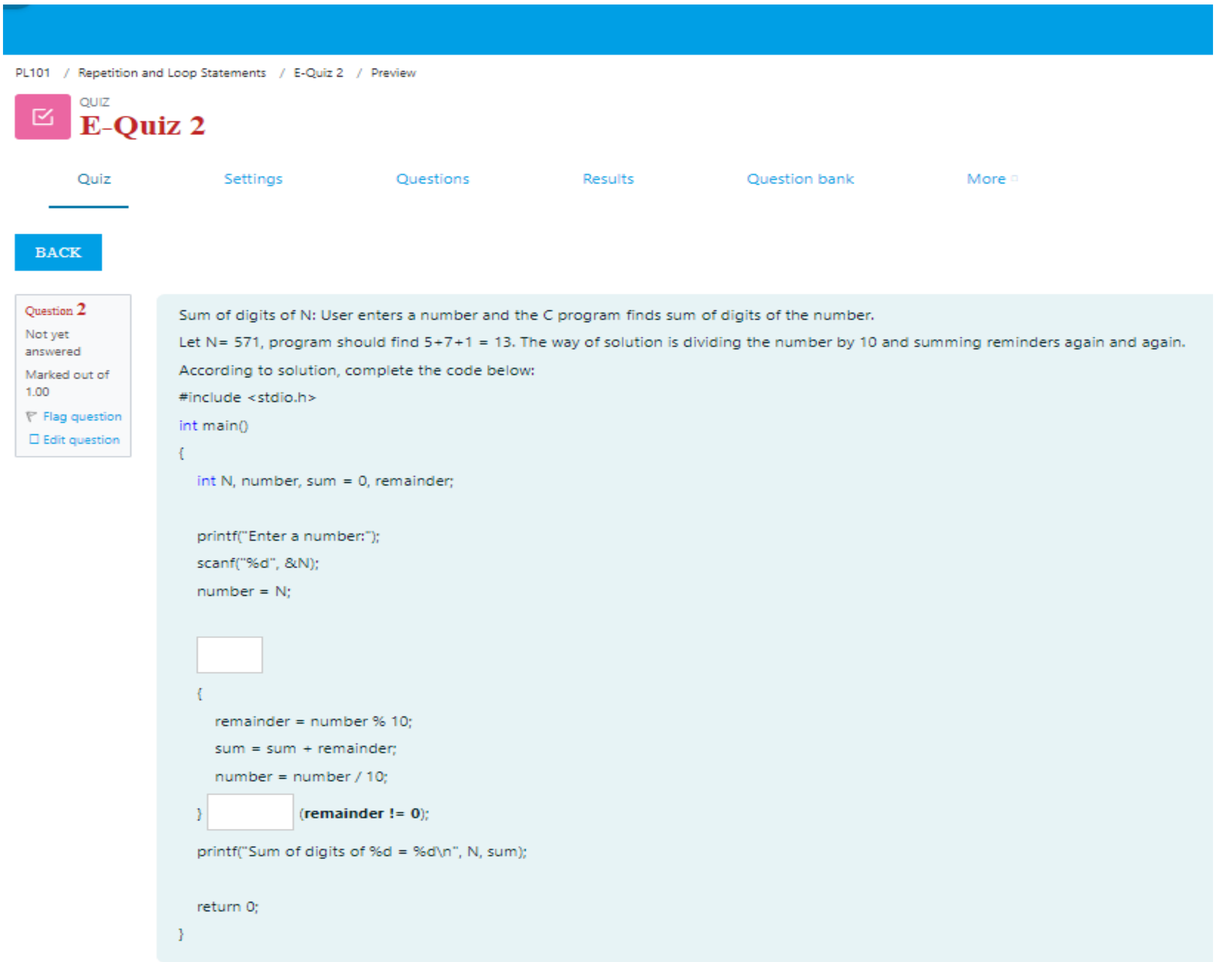

Figure 5: A sample e-quiz question interface on web-platform

Moodle supports various content types. For example, it has a built-in video player which can play local files, streams, or external content from different sources such as YouTube videos. Besides, the contents can be enriched with H5P, SCORM packages, and others by using plugins. Interactive content developed using various tools such as game engines and other plugins can be published as SCORM packages or H5P. Figure 6 shows a game developed in the Unity game engine, deployed to WebGL, and exported as a SCORM package.

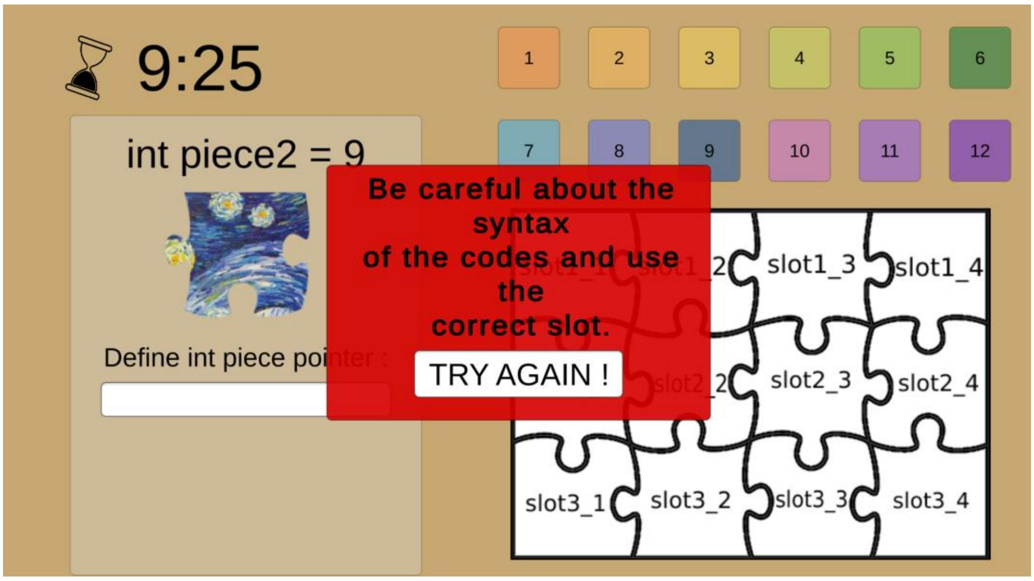

Figure 6: A game developed as SCORM package

3.2. The Educational Materials

The educational materials are developed in line with their design as outlined in the course plan and integrated into the web-platform. The educational materials consist of interactive video tutorials, mobile games, e-quizzes, and case studies. In the course design, each module has at least two different types of those educational resources. In this way, student-centred learning is provided by different types of learning activities.

Video tutorials: We prepared a series of interactive video tutorials to teach various topics in computer programming. Namely, each of Topics 1, 4, 7, 9, and 10 - given in Table 1 includes one video tutorial. First, we created a scenario for each game and then created their storyboards accordingly. Each video tutorial covers real-life examples of programming languages.

Mobile games: We prepared mobile-compatible games. Each game has a specific theme in relation to the topic of the module. First, we created a scenario for each game and then created their storyboards accordingly. In total, we designed and developed five games for Topics 2, 3, 6, 8, and 9 - given in Table 1.

E-quizzes: For Topics 2, 5, 6, 7, and 10 - given in Table 1, we prepared 3–5-minute surprise questions to motivate students to do the readings. We prepared scenarios for each quiz, and designed storyboards accordingly. The quizzes consist of multiple choice, fill-in-the-blanks, and short answer type of questions.

Case studies: For Topics 2, 4, 7, 8, and 10 - given in Table 1, we prepared case studies to make students apply course knowledge to produce something new. Each case study focuses on a specific algorithm problem and also includes a solution tip for students. For each case study, we prepared a scenario that contains multiple steps showing the general definition and problem statement to enable students to better understand the case. For each step, we defined action, dialogue/script, and interaction/navigation.

4. Conclusions

In sum, this paper presents a case study to contribute to the online/hybrid learning environments for programming courses in engineering faculties. The case study is structured within the framework of the Kemp Instructional Design model to construct an instructional design. The instructional design includes a course plan, a web-platform, and related educational resources - the main outputs of the RECOM project.

Formerly, as the project team, we have already developed a course design template (**Uvet et al. 2022**), which is based on the need analysis, as the Kemp ID model suggested. The template is mainly characterized by instructional strategy issues closely related to the student-centred learning approach, such as feedback, gamification, and the flipped classroom. Its development has been preceded by a needs analysis, consisting of analyses of instructional problem, learner characteristics, content, and instructional objectives. Accordingly, the course content has been planned to include various kinds of educational resources: interactive video tutorials, mobile games, e-quizzes, and case studies. To deliver the course content, an environment has been decided upon as a web-platform that also has mobile access.

Using this template, we created the instructional design. It is scaffolded by a learner centred course design template and consists of educational materials integrated into the web-platform that is built on Moodle. All the efforts point out the ultimate goal of the project, which is to offer an effective online/hybrid learning environment for programming language courses for engineering students. It is a holistic approach with both the pedagogical aspects of the learning/teaching process and the technological aspects with the appropriate educational technology materials and web-platform. Therefore, it will have benefits for teachers to have adequate pedagogy to deliver online/hybrid programming courses effectively, either synchronously or asynchronously. In this way, students will have well-structured online programming courses with individualized instruction where students can have opportunities to analyze and assess their own progress within their course.

For the rest of the project's duration, we aim to evaluate the course design with all the educational materials integrated into the web-platform. It will be accomplished after the user testing process. This process will include piloting with target group representatives from all the partner countries. This piloting will start in person during learning/teaching/training activities and will continue with hybrid piloting in all the partner countries. The feedback from all the participants will be analysed with the purpose of improving the course design according to the teacher's/students recommendations. This feedback could be further extended according to the Kirkpatrick model (**Kirkpatrick 2006**), which consists of four evaluation levels, namely Reaction, Learning, Behaviour and Results.

Future work may also include the examination of the framework's applicability in different programming-related subjects, like different languages, technologies like cloud computing, Internet of Things, security, networks, etc.

References

Aberdour, Mark. 2013. Moodle for Mobile Learning. Packt. https://www.packtpub.com/product/moodle-for-mobile-learning/9781782164388.

Akcil, Umut, and Mert Bastas. 2020. "Examination of University Students' Attitudes towards E-Learning during the COVID-19 Pandemic Process and the Relationship of Digital Citizenship." Contemporary Educational Technology 13 (1): ep291. https://doi.org/10.30935/cedtech/9341.

Bybee, Rodger W. 2010. "Advancing STEM Education: A 2020 Vision." Technology and Engineering Teacher 70 (1): 30–35.

Davies, Randall S., Douglas L. Dean, and Nick Ball. 2013. "Flipping the Classroom and Instructional Technology Integration in a College-Level Information Systems Spreadsheet Course." Educational Technology Research and Development 61 (4): 563–80. https://doi.org/10.1007/s11423-013-9305-6.

Dichev, Christo, and Darina Dicheva. 2017. "Gamifying Education: What Is Known, What Is Believed and What Remains Uncertain: A Critical Review." International Journal of Educational Technology in Higher Education 14 (1): 9. https://doi.org/10.1186/s41239-017-0042-5.

Förster, Anna, Jens Dede, Asanga Udugama, Alexander Förster, Daniel Helms, Louis Kniefs, Julia Müller, Lars Gerken, Franziska Richter, and Jan Kulmann. 2021. "A Blended Learning Approach for an Introductory Computer Science Course." Education Sciences 11 (8): 372. https://doi.org/10.3390/educsci11080372.

Hampsten, Katherine. 2021. "Embracing Discomfort and Resisting a Return to 'the Good Old Days': A Call to Communication Educators." Communication Education 70 (2): 208–10. https://doi.org/10.1080/03634523.2020.1857413.

Hehir, Elizabeth, Marc Zeller, Joanna Luckhurst, and Tara Chandler. 2021. "Developing Student Connectedness under Remote Learning Using Digital Resources: A Systematic Review." Education and Information Technologies 26 (5): 6531–48. https://doi.org/10.1007/s10639-021-10577-1.

Hu, Yiyang, Siyuan Sang, and Chen Meng. 2021. "Reviewing the Interactions Between Instructors and Students in Online Education:" In . Hangzhou, China. https://doi.org/10.2991/assehr.k.210824.020.

Jaftha, Nevena, Marouska Zahra-Micallef, and Tatjana Chircop. 2021. "The Impact of Gamified Instruction on Students' Learning Outcomes: Systematic Review of Experimental Studies." International Journal of Education 13 (4): 55. https://doi.org/10.5296/ije.v13i4.19193.

Kirkpatrick, Donald L., and James D. Kirkpatrick. 2006. Evaluating Training Programs: The Four Levels. 3rd edition. San Francisco, CA: Berrett-Koehler Publishers.

Kumar, Sheo, Anil Kumar Gankotiya, and Kamlesh Dutta. 2011. "A Comparative Study of Moodle with Other E-Learning Systems." In 2011 3rd International Conference on Electronics Computer Technology, 414–18. Kanyakumari, India: IEEE. https://doi.org/10.1109/ICECTECH.2011.5942032.

"About Moodle - MoodleDocs." n.d. Accessed December 19, 2022. https://docs.moodle.org/400/en/About_Moodle.

"Moodle Statistics." n.d. Accessed December 19, 2022. https://stats.moodle.org/.

Morrison, Gary R., Steven M. Ross, Howard Kalman, and Jerrold E. Kemp. 2010. Designing Effective Instruction. 6th edition. Hoboken, NJ: John Wiley & Sons.

Nortvig, Anne-Mette, Anne Kristine Petersen, and Søren Hattesen Balle. 2018. "A Literature Review of the Factors Influencing E-Learning and Blended Learning in Relation to Learning Outcome, Student Satisfaction and Engagement." Electronic Journal of E-Learning 16 (1): 46–55. https://eric.ed.gov/?id=EJ1175336.

Omeh, Christian B., Chijioke J. Olelewe, and Emmanuel C. Nwangwu. 2022. "Impact of Teaching Computer Programming Using Innovative Pedagogy Embedded with Live Online Lectures and Related Tools: A Randomized Control Trial." Computer Applications in Engineering Education 30 (5): 1390–1405. https://doi.org/10.1002/cae.22527.

Persano Adorno, Dominique. 2022. "Inquiry-Based Environments for Bio-Signal Processing Training in Engineering Education." International Journal of Mechanical Engineering Education 50 (3): 629–47.https://doi.org/10.1177/03064190211026207.

Rapanta, Chrysi, Luca Botturi, Peter Goodyear, Lourdes Guàrdia, and Marguerite Koole. 2020. "Online University Teaching During and After the Covid-19 Crisis: Refocusing Teacher Presence and Learning Activity." Postdigital Science and Education 2 (3): 923–45. https://doi.org/10.1007/s42438-020-00155-y.

Rincon-Flores, Elvira G., and Brenda N. Santos-Guevara. 2021. "Gamification during Covid-19: Promoting Active Learning and Motivation in Higher Education." Australasian Journal of Educational Technology 37 (5): 43–60. https://doi.org/10.14742/ajet.7157.

Rossi, Izadora Volpato, Jordana Dinorá Lima, Bruna Sabatke, Maria Alice Ferreira Nunes, Graciela Evans Ramirez, and Marcel Ivan Ramirez. 2021. "Active Learning Tools Improve the Learning Outcomes, Scientific Attitude, and Critical Thinking in Higher Education: Experiences in an Online Course during the COVID -19 Pandemic." Biochemistry and Molecular Biology Education 49 (6): 888–903. https://doi.org/10.1002/bmb.21574.

Saleem, Awaz Naaman, Narmin Mohammed Noori, and Fezile Ozdamli. 2022. "Gamification Applications in E-Learning: A Literature Review." Technology, Knowledge and Learning 27 (1): 139–59. https://doi.org/10.1007/s10758-020-09487-x.

Singh, Jitendra, Keely Steele, and Lovely Singh. 2021. "Combining the Best of Online and Face-to-Face Learning: Hybrid and Blended Learning Approach for COVID-19, Post Vaccine, &

Post-Pandemic World." Journal of Educational Technology Systems 50 (2): 140–71. https://doi.org/10.1177/00472395211047865.

Strayer, Jeremy F. 2012. "How Learning in an Inverted Classroom Influences Cooperation, Innovation and Task Orientation." Learning Environments Research 15 (2): 171–93. https://doi.org/10.1007/s10984-012-9108-4.

Trammell, Beth A, and Chera LaForge. n.d. "Common Challenges for Instructors in Large Online Courses: Strategies to Mitigate Student and Instructor Frustration." Journal of Educators Online Educators Online 14 (1): 1–10. https://files.eric.ed.gov/fulltext/EJ1133615.pdf.

Uvet, Huseyin, Tuba Ugras, James S. Quaicoe, Abiodun Ogunyemi, Merja Bauters, Dominique Persano Adorno, Veselina Jecheva et al. 2022. "Designing for Student-Centered Hybrid Learning Environments: A Framework for Programming Languages Course Design." InHELMeTO 2022, Book of Abstract, 104-106. Palermo, Italy. https://drive.google.com/file/d/1d4OYdCARW-tCs85IQULU9stVWUjcJyTp/view.

Vargas-Macías, Zulma Liliana, Ariel Adolfo Rodríguez-Hernández, and Claudia Liliana Sánchez-Saenz. 2020. "Digital Games (Gamification) in Learning and Training: An Approach to Adaptation and Integration in the Classroom." GIST – Education and Learning Research Journal 20 (June): 171–88. https://doi.org/10.26817/16925777.765.

Acknowledgments

The study has been conducted with the support of the Erasmus+ Programme of the European Union, Project ID 2020-1-TR01-KA226-HE-098258.