JOURNAL OF E-LEARNING AND KNOWLEDGE SOCIETY
Vol. 19, No. 1 (2023), pp. 27-35

Online tutoring system for programming courses to improve exam pass rate

Giacomo Nalli[a] , Rosario Culmone[a], Andrea Perali[b], Daniela Amendola[c]

[a]*University of Camerino, School of Science and Technology, Computer Science Division – Camerino (MC), Italy*

[b]*University of Camerino, School of Pharmacy, Physics Unit – Camerino (MC), Italy*

[c]*University of Camerino, School of Biosciences and Veterinary Medicine – Camerino (MC), Italy*

(submitted: 20/6/2022; accepted: 24/4/2023; published: 27/4/2023)

Abstract

University students enrolled in the first year of the Computer Science degree may have problems approaching programming, negatively affecting their study during the course. Tutoring programming projects are very important in helping students with difficulty in learning by providing the right approach to study, improving their knowledge and skills in computing. The aim of this work is to realize a new Java Programming tutoring online course that allows students to have an effective online tool to achieve the learning goals of the course and this will enhance the programming exam pass rate. The course we have designed consists of tools to help students with video tutorials, self-assessment quizzes, code evaluations and exercises to solve using an online Java editor. Because the Moodle platform lacks tools to check the quality of the code syntax, a new software was created. It performs a syntax analysis of the Java code and, as a tutor, automatically provides feedbacks and tips to the students to improve the quality. For each online tool the immediate feedback technique is used to amplify students' engagement. A Clustering Machine Learning technique is performed to identify different students' behaviors. A correlation between them and the final performance showed the most influential features of the completed activities. Quantitative analysis highlighted the effectiveness of the tutoring system and the online course designed in this work to enhance the final exam pass rate. At the end, students filled a questionnaire to report their perception and satisfaction about the course.

KEYWORDS: Tutoring, Feedback, Java Programming, Moodle, Machine Learning.

DOI
https://doi.org/10.20368/1971-8829/1135704

CITE AS
Nalli, G., Culmone, R., Perali, A., & Amendola, D. (2023). Online tutoring system for programming courses to improve exam pass rate. *Journal of e-Learning and Knowledge Society*, *19*(1), 27-35. https://doi.org/10.20368/1971-8829/1135704

1. Introduction

In the first year at university, starting a new programming course can be hard for computer science students. Sometimes they can also have difficulty to pass the exam. Programming courses are perceived as difficult. This can increase the student dropout rate (Robins et al., 2010) and can negatively affect student study (Ambrosio et al., 2011; Aşkar et al., 2009; Figueredo et al., 2020; Hawi, 2010). A lot of the students that don't have any programming experience before, think that programming is difficult to learn and use. They refuse to learn programming skills unconsciously (Tan et al., 2009). Students' views regard the difficulties that they faced in programming themes that cover semantics, programming skills and programming knowledge (Özmen et al., 2014). Many students had difficulties developing computational thinking. Computational thinking is an attitude that permits problems solving using computational techniques (Wing, 2006; García-Peñalvo, 2016). Recent studies identify teaching methodology as the main reason for problems in teaching and learning programming. They rarely focused on developing the skills and knowledge acquired in the student's learning experience (Figueiredo et al., 2020).

Students must study not only theoretical concepts, but also interpret their own mistakes, find examples, and search solutions to known problems. The tutorial aspect

is important because it can allow the improvement of the quality of teaching. In fact, it can drive the student to acquire generic and disciplinary skills. Tutoring represents an action to support the different aspects of a student's learning development (De Santis et al., 2021). It helps students during their study and supports them in any difficult moments, such as the new approach to university study. The purpose of tutoring is to provide students to approach the critical steps of their studies in a good way. It is aimed at promoting academic success and avoiding university drop-out (Da Re, 2018). Sometimes it can happen that the degree courses in Computer Science are composed of many students enrolled. IT resources are often used to have a good management of classrooms with a lot of students. These resources are very effective because they allow to improve learning processes, thanks to new pedagogical approaches like e-learning courses applied in the field of Computer Science. Indeed, the use of MOOC for teaching IT in a university course with a high number of students, showed how these teaching methods have been effective, and expanded the students' computer knowledge and skills. Furthermore, the use of the online course did not reduce the teaching effectiveness and allowed the improvement of the student's learning results (Amendola et al., 2018; Guelfi et al., 2020). Blended programming e-learning courses provided some help for students to gain practical experience through the involvement of Java programming activities. They allowed students to learn more easily and are adapted to acquire programming skills (Hadjerrouit, 2008).

Virtual Laboratories are also provided in e-learning for learning techniques and knowledge relating to computer programming. These online labs provide hands-on simulations that permit the students to learn comprehensively, through face-to-face lessons. (Richter et al., 2012). In online laboratories for learning graphic programming software such as Gnuplot, Video Tutorials are often used for data collection applied to the code (Amendola et al., 2016). Given the success of e-learning courses in education, its effectiveness for student learning, and the difficulty of managing a lot of students in the classroom, a programming tutoring e-learning course is created.

An online programming tutoring course can be very useful because e-learning platforms, like Moodle, allow the integration of different tools for programming purposes. Furthermore, they give the possibility to set for each tool automatic feedback. Feedback helps students to understand both the learning outcomes of the course and their relative progress toward meeting those goals (Hattie et al., 2007). Research evidence shows how effective feedback is an important element for student learning (Shute, 2008). Feedback finds many applications in the programming field, and it is used to contribute to improving students' cognitive outcomes (Gusukuma et al., 2018; Marwan et al., 2019). One of the fundamentals criteria that affect efficacy in feedback is the timing. Immediate feedback gives a lot of advantages, like the promoting of actively learning. It allows students' partial knowledge to be rewarded with partial credit and it is strongly preferred by students compared to other techniques (Di Battista, 2005). An example of immediate feedback is the IF-AT that provide to students' immediate feedbacks about the accuracy of their answers, as the students are completing each exercise. The Immediate Feedback Assessment Technique (IF-AT) system provides immediate affirmative feedback or corrective feedback, depending on the answer given by the students (Epstein et al., 2002). Research evidence shows how immediate feedback is more effective compare the delays feedback, on complex tasks, when students have less prior knowledge (Shute, 2008) like the new university students that approach programming. Students involved in immediate feedback got a high level of scores and correct answers then the other students. This method engages students in the process of discovering and correcting initially imprecise response strategies (Epstein et al., 2002).

Learning Management System (LMS) like Moodle permits to create an effective programming tutoring e-learning course, because it covers most of the features that allow the successful learning of java programming, like videos, online editors, and different types of multiple-choice quizzes. These tools allow the integration of video tutorials for writing code, give the possibility to perform the code and, thanks to the feedback functions, check errors in the execution of the created software, check theoretical knowledge, evaluate the code, test the functionality of the code. However, testing the quality of the syntax produced by the student is important to complete an online Programming Tutoring course.

An online editor can recognize errors at runtime; anyway, if the program runs correctly, the code may not be written optimally. It could not respect the logic of the object-oriented paradigm, or there could be an improper use of java commands that could create optimization problems. Unfortunately, in Moodle there is no automatic tool that allows the students, based on the code described, to have feedback and advice to improve the quality of the code syntax of the Java programming language. A new tool was created to analyse the syntax of the written code and its quality. The tool then, as a tutor, automatically provides feedback and tips to the students in order to improve the quality of the code.

In recent years, several students of Computer Science at the University of Camerino encountered difficulties in Java programming during the attendance of their first-year degree. These difficulties were not always overcome, and students often didn't pass the exam during their first-year degree, registering a decreasing trend of exam pass rate from 51% to 43% during the

last four years (from academic year 2015/2016 to 2018/2019). The Java programming course is one of the most important courses in the three-year degree in Computer Science, so the exam failure in the first year led often students to leave the university. We recall here that the first-year drop out from the university system is one of the most crucial problem in Italy and one of the most difficult to be solved, or at least mitigated.

The aim of this work is to realize an effective online Tutoring course that contributes to an improvement in the quality of teaching in the Java Programming course.

The research conducted aims at investigating if the online tutoring system is able to improve the students' performance and also enhance the exam pass rate in the first-year students in order to decrease the possibility of drop-out.

The final goal of our work is to demonstrate how it improves students' computer skills and knowledge and check if they can give benefits to students, in terms of performance (if they quickly pass the final exam) and in terms of satisfactions (analysing also the questionnaire based on the students' perceptions).

The online tutoring course was delivered in January 2020 and was structured in 6 modules, designed, and uploaded to the Moodle platform of the University of Camerino. Students of the first year of the degree course in Computer Science had access to the course at the end of the period in which the Java Programming lectures were held. They performed the activities of the tutoring course before the final exam. In this paper, the design of the online course and the results obtained by the students were discussed to evaluate the impact of the online tutoring course in terms of pass rate of the final exam and level of student satisfaction. In fact, we want to evaluate the effectiveness of the online tutoring course, to ensure an effective tool for improving the study approach to students.

The evaluation the course was analysed by the following research questions:

1. How did the students use the online resources?
2. Did the massive use of the platform help students pass the exam within the academic year?
3. How did the students perceive the effect of online resources in acquiring skills and passing the exam?

A teaching model was designed and adopted to obtain the answers to these research questions. Logs for the various activities were extracted from the e-learning platform and processed using Machine learning techniques. In this way it was possible to determine and compare the behaviours held by the students on the platform and how they affected the results of the final exam. Finally, the questionnaire filled by students was analysed.

2. Methods

2.1 Participants

This case study was carried out with a group of 151 first-year students of the degree course in Computer Science at the University of Camerino. The course was delivered in Italian. Participation in the online tutoring course was voluntary. The online tutoring course was available to students after the theoretical and laboratory face to face lectures, to have a basic preparation before taking the tutoring course. A professor, a tutor, and a Ph.D. student in Computer Science course at the University of Camerino prepared the teaching materials, the video tutorial and designed the online activities.

2.2 Research methods and procedures

The project of the online tutoring course was preceded by an accurate identification of the topics that the last years students found difficulties.

The main goal of teaching programming course is to acquire skills to create computer software that solve real problems, developing computational skills. The course was divided into 6 modules, each of which consists of topics that allow to achieve the learning outcomes of the programming course. These permit to implement and create simple software to solve computational problems. In particular, the following topics were explained in each module:

1. primitive data types, cast, bitwise operators;
2. vectors and cycles;
3. control flows and logical operators;
4. classes and exceptions;
5. math and Util libraries;
6. inheritance and interfaces.

For each module, activities were developed to allow effective learning of students' knowledge and skills. Each module initially consists of a video tutorial, often used in e-learning courses to stimulate student engagement (Amendola et al., 2016). The video tutorial drive students step by step in writing the programming code aimed at solving programming exercises. Video tutorials last a maximum of 20 minutes, about 40 MB in size and consist of voice and screen recording. In particular, a Java online editor is displayed, and the step-by-step explanations show the code to solve exercises. Key information on the methods and solutions are used; theoretical references are included in the video, to provide the tools to solve similar tasks. To give the opportunity to develop computational knowledge, various activities and exercises are included where students with difficulties can test their skills. The activities to be delivered in the course are based on teaching models that found a fast and effective learning of computer skills and knowledge in programming (Tan et al., 2014). These models consist

of different exercises like multiple choice and open cloze. The interactive tutor is another important tool for learning success which supports step by step the development of simple computer software (Figueiredo et al., 2020).

The following activities are included in each module:

1. multiple choice quiz;
2. text completion;
3. code evaluation;
4. exercise with solution;
5. programming exercise.

The "multiple choice quiz" consists of written java code and student must give the right answer selecting what output the code performs.

The "text completion" is composed by open cloze exercises, used for the acquisition of knowledge related to the Java syntax (e.g., logical operators, conditional operators, data types etc.).

The "code evaluation" is a multiple-choice quiz that, analysing a written code, consists of more than one alternative that students must detect. In this exercise there is the code that has anomalies. Students have to check what anomalies it presents by selecting the correct ones from a series of items. The system automatically gives feedback showing the theoretical explanation.

In the "exercise with solution" activity there is a task that describe the problem to perform and the solution, giving the complete source code. This exercise is useful for students because they can try to solve the task by themselves and later compare with the solution provided. In this way they can understand the correct way on how to solve the problem properly.

Like this activity, there is a "programming exercise" that students have to perform autonomously. Students must produce the source code on the java editor and check if it's correct.

The Java editor used in this course is an online compiler made by "Trinket" that is embedded into the Moodle course. It runs the java code online and checks its correct execution.

When the source code is correct, students check the quality of the syntax code. The students execute the implement code using the "interactive tutoring software", that we especially developed for this online tutoring course to overcome Moodle limits (Moodle lacks a tool that supports these features). This software provides automatically immediate feedback on the quality of the code entered, analysing criteria like "keywords", "use of variables", "use of methods", "correct use of classes" to satisfy the object-oriented paradigms.

There are several studies that highlight the effectiveness of an interactive tutor in teaching computer science: i) working with an interactive tutor who supports the making of programs is more effective than learning to program by doing the same exercise using only a compiler; ii) the use of automatic tutors requires less help from the teacher; iii) use of online tutors increases self-confidence in students; iv) immediate feedback from the tutor appears to be preferable to feedback given later in class (Gerdes et al., 2012). The interactive tutor application is developed in PHP and is installed on a Linux, Apache, MySQL e PHP (LAMP) server. It consists of an html input file, which consists of a textbox where students insert their own java code and a button for code verification and an output php file, which shows to students feedback related to the inserted code. Students receive, on output, feedback on mistakes and tips to improve the quality of the java syntax. The system consists of an accurate analysis of the text that it takes as input. It compares the written text with different criteria that the code should satisfy, set by the teacher for a given exercise. These criteria can be: i) use of specific keywords (e.g. "final", "static", "public"), ii) length of the code, iii) use of java classes, methods, libraries and functions.

Thanks to these tips, students can understand their mistakes and edit their code until they get positive feedback, improving their code without the human intervention. A database, connected to the interactive tutor, collects all the students' attempts every time they try to verify the quality of the written Java code. Thanks to this mechanism the teacher, who has access to the database, can directly see each attempt and check, based on the students' mistakes, if there are topics that are difficult for the students to understand. Monitoring the student progress checking the data collected by the tool is an important feedback for the teacher which can detect possible issues in specific topics and in case edit properly the online activities or learning materials in order to guide the students to overcome their weakness in these topics.

In addition to the 6 modules, the course has an introductory video that explains: i) aims and structure of the course; ii) how to use the editor; iii) general intro to java topics.

Finally there is a questionnaire, consisting of 24 questions aiming to acquire interest in the study and to check students' perception and satisfaction.

2.3 Data collection and methodology

The results consisted of an analysis of student reports and a questionnaire on the use of the course, both extracted by Moodle platform.

Several aspects were analyzed to determine the effectiveness of the Programming tutoring course:

1. use of the platform activities, through online course modules logs;
2. detection of the percentage of students who passed the exam based on their behavior in the online programming tutoring course;

3. data collected from an online questionnaire aiming to extract info about students' experience and perceptions in the programming tutoring course.

Initially (point 1) we analyzed the learning analytics extracted by the e-learning platform at the end of the tutoring course. These data were organized in tables showing the number of logs to the different resources and activities, to determine the level of interaction of the students.

Then (point 2) we checked how an intensive use of the platform can increase the success rate for passing the final exam. To do this, we first determined the different students' behaviors based on the display and the use of all the activities in the modules. In online learning environments, clustering can be used to find groups of students with similar behavior and characteristics (Vellido et al., 2010; Nalli et al., 2022). Clustering is a Machine Learning technique that permits automatically to identify relationships between data in a dataset with multiple features, to group objects with similar characteristics in the same group(cluster) (Bovo et al., 2013, Nalli et al., 2021). The clusters are generated by: i) extraction of data relating to the students from the Moodle Reports used to create the dataset, ii) the input file of the clustering algorithm; iii) creation of the software to execute the clustering algorithm to grouping students with similar characteristics and behaviors.

We selected the following data relating to the behavior of the students: total time and login frequency, number and attempts of completion exercises, number and attempts of evaluation code exercises, number and attempts of quizzes, number and attempts of exercises with java compiler and interactive tutor, number and frequency of videos. To simplify datasets, we grouped data into a few features. The feature is an individual and measurable property of an observed phenomenon (Bishop, 2006). Data belonging to the same type of activity (but also in different modules) was aggregated into the same feature.

The collected data was preprocessed. Each student was represented by a "vector of input" with features consisting of the values of the attributes associated with the student.

All the data, organized into feature vectors, one for each student, were inserted into a single Excel file, called dataset, which represents the input file of our clustering software.

The clustering software with K-Means algorithm was implemented through Python scikit learn library (Hackeling, 2014).

The software creates groups of students that have similar behaviours and characteristics on the platform.

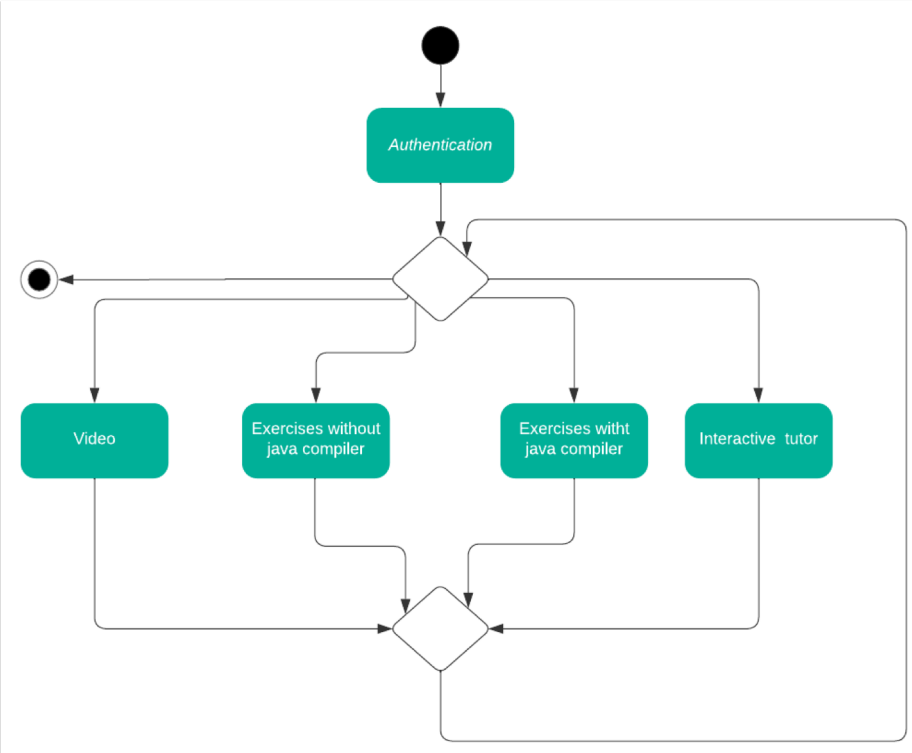

Figure 1 – Conceptual flow of online activities and software execution.

Once obtained the clusters, we compared them with the percentages of students who passed the final exam before the end of the academic year 2019/2020; thus determining the most influential behaviours and activities that allowed a fast pass exam rate.

Finally (point 3), the course provided a questionnaire consisting of 24 questions divided into 3 sections: i) behaviours, ii) intentions, preferences, opinions, iii) open questions and comments. The section "behaviours" consists of 12 questions to obtain data on the experience related to the online tutoring course. The section "intentions, preferences, opinions" consists of 9 questions, using Likert scale. These questions covered satisfaction and perception related to the online course, in particular advantages and disadvantages related to the understanding of the material and use. Section "open questions and comments" consists of 3 open questions, where students give their opinion on what they liked most and what they liked least about the course, useful as feedback. The students completed the questionnaire voluntarily and anonymously. In this paper we discuss the results related to the second section of the questionnaire to detect the students' perception and satisfaction.

3. Results

How did the students use the online resources?

The data relating to students enrolled in the first year of the Computer Science course of the academic year 2019/2020 were analysed.

Table 1 summarizes the level of general interaction that the students had with the materials within the different modules. For each user, multiple logs occurred for the same activity.

Access to the activities of each module was unrestricted and therefore students didn't need to follow an order for viewing the materials. Therefore, students only viewed the activities in which they had lot of difficulties or the activities useful for passing the exam.

Table 2 shows the types of activities most used in the course. From the number of accesses, we can see how the practical activities, which return an evaluation or feedback, had more success than the theoretical activities (such as videos). This happened because the students enrolled in the tutoring course needed to test their knowledge through exercises, compared to the theory carried out on face-to-face lessons. All exercises were delivered on the online course at the same time and students were able to perform the exercises with an unlimited number of attempts. The most used activity was the "Multiple choice quiz" as it is an important tool for students to fix their knowledge. Other activities widely used were "Text Completion" and "Code Evaluation", which return an immediate evaluation based on the option selected or word entered. The possibility of having immediate feedback allows students to check if they understood the concepts, or to identify errors with the aim of improving their programming knowledge. The "programming exercise" activity was also widely used, with 1331 accesses, which was preferred to "Exercise with solution". This reinforces the importance of the interactive tutor to learn programming and develop computational knowledge. Using the online tutor with immediate feedback, the student is driven in writing the code with a good quality. This allows students to be more effective in learning programming than carrying out this activity by themselves or using only the compiler; it also improves their self-confidence. (Figueiredo et al., 2018).

Materials	Logs
Module 1 - Primitive data types, cast, bitwise operators	4672
Module 2 - Vectors and cycles	3708
Module 3 - Flow control and logical operators	1891
Module 4 - Classes and exceptions	2042
Module 5 - Math and Util libraries	1708
Module 6 - Inheritance and interfaces	1433

Table 1 – Number of student-logs to the materials in all Modules of the online course.

Activities	Logs
Video	987
Multiple choice quiz	8391
Text completion	2057
Code evaluation	2061
Exercise with solution	639
Programming exercise	1331

Table 2 – Most used activities in the online course.

Has massive use of the platform helps students to pass the exam within the academic year?

To determine the effectiveness of the online tutoring course, we decided to carry out a quantitative analysis on the dataset for the first-year students of the Computer Science Degree Course of the academic year 2019/2020. Initially we used the Clustering technique to determine groups of students who had similar characteristics relating to the use of the course activities. The second step was the correlation of each cluster with the performance obtained by students belonging to the same cluster, to determine which behaviours influenced passing the exam. Only exams passed within the 2019/2020 academic year were analysed.

The clusters were examined to determine the different behaviours on the platform, highlighting which features are the most influential. In this way it was possible to profile the students who belong to the different clusters. Cluster 0 (26 users) represents students with high activity in the course, Cluster 1 (40 users) represents students with average activity and Cluster 2 (85 users) reflects inactive students. This is also confirmed by the analysis of very influential features in the creation of clusters, using the K-means clustering algorithm.

The most important features were the following: *i*) number of quiz attempts, *ii*) number of views of the example performed and number of exercises to be performed with the compiler and interactive tutor.

The feature values within each cluster are represented using the average of the feature values of the students belonging to the same cluster. The average value of each feature is shown in comparison with the maximum value that the feature can represent.

Cluster 2, that represents the inactive students, showed few attempts among all the quizzes of the course with an average of 7.37 attempts, few views of the types of "exercises with solution" (0.42 / 6) and a low number of different exercises carried out with java compiler and interactive tutor (0,81 / 12). This little interaction during the course is also confirmed by the number of accesses of the students belonging to the cluster (3.42) and by the number of video views (1.8).

Students belonging to cluster 1 reflect average values in all activities. In fact, Cluster 1, compared to Cluster 2, finds a significant difference such as the number of quiz attempts (80.52), number of views of exercises with solutions (2.52 / 6) and number of different exercises carried out with java compiler and interactive tutor (4.62 / 12). The number of accesses (16.12) confirms the greater activity of these students and the video views (12.02).

Cluster 0, that represent active students, showed an high number of quiz attempts (130.26), lot of views of exercises performed (4.84 / 6) and an high number of exercises performed with java compiler and tutor (10, 23 / 12).

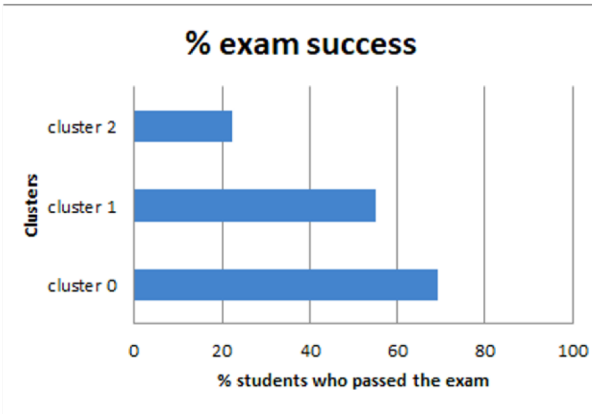

Figure 2 – Number of students per cluster who passed the final exam within the academic year 2019/2020.

To check if the behaviour carried out by students on the e-learning platform had benefits in terms of performance, we extracted from each cluster the percentage of students who passed the final exam before the end of the academic year 2019/2020.

As showed in Figure 2, the students of Cluster 0, with high activity on the platform, achieved an higher success rate than the other clusters, equal to 69%. A good success rate was returned also by the students of Cluster 1 who, with average activity, passed the exam with a percentage of 55%. Cluster 2 had a very low percentage of students who passed the final exam, equal to 22%. It confirms the tutoring course gives benefits for the students.

How did students perceive the effect of online resources in acquiring skills and passing the exam?

In this part we reported the results of the data returned by the questionnaire proposed to first year students at the end of the last exam session of the academic year 2019/2020. We wanted to analyse the perception of students on the use of the online tutoring course, and especially if this course influenced the improvement of computational knowledge and skills in addition to passing the final exam.

Most of the students believe that the use of the online tutoring course is effective for passing the exam. Figure 3 shows that 76% of students believe in the usefulness of the tools and exercises, to increase the chances of success, while only 23% think that they had little or no impact. This last part of students was represented by them that had less activity online. They, as shown by the Clusters analysis, didn't easily pass the exam.

The usefulness of the tutoring course is also confirmed in the improvement of computational skills and knowledge. 80% of students claim that the tools and exercises used in the course have improved their computational skills, while 20% believe that they had

little influence on learning skills. Regarding the improvement of the knowledge, 85% of students believe that the tools improved them, compared to 15% of students who think that they had little impact for the knowledge enhancement. 95% of students consider all online activities (Text completion, Code Evaluation, Exercise with solution, Programming exercise) useful as a teaching support tool.

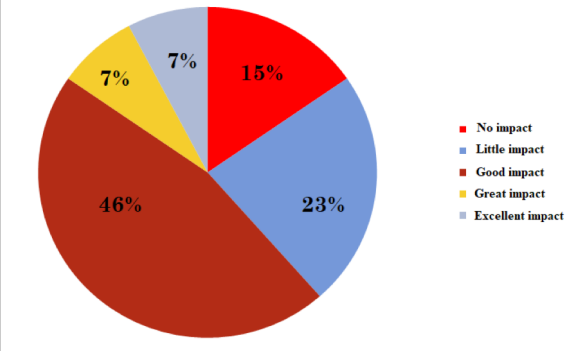

Figure 3 –Impact that the tools and exercises in the online course designed in this work had on passing the exam within the academic year 2019/2020.

The video tutorials were appreciated as support tools, with only positive judgments (50% = good, 28% = very good, 22% = excellent). This result is confirmed by the students that consider video tutorials positively for the improvement of knowledge (89%) and skills (83%). However, only 57% consider the video tutorials useful for passing the exam, while 43% consider them not very useful. This highlights the need for students to have tools that allow them to practice computation, compared to theoretical notions.

4. Conclusions

In this work, an online Java Programming tutoring course was created and delivered through the Moodle platform, which allowed the students with learning difficulties to have an effective online tool to improve their computational skills and knowledge, overcoming the weakness and critical phases of their studies.

This was also possible thanks to the possibility of using video tutorials and carrying out specific online programming exercises.

Due to the lack of a tool to check the syntax quality of the code written by the students, we implemented a new software to achieve this goal. We therefore created a new Tool (Interactive Tutor) which performed a syntax analysis of the written code and, as a tutor, automatically provided feedbacks and advice to improve quality. This tool doesn't require the teacher intervention and speeds up the learning process.

The results reported in this work showed the effectiveness of the online tutoring course and a high level of students' engagement. The comparison between the clusters obtained and the measured success rate highlighted the impact that the course had in terms of passing the exam. Cluster 0, which consists of active students in the course, had a high success rate compared to the others equal to 69%. Cluster 1, that involved students with an average activity, had a lower exam pass rate of 55%, while Cluster 2, which includes students with low activity, had a percentage of 22% exam pass rate.

The quantitative analysis, carried out by processing the data extracted from the final questionnaire filled by the students, reported excellent feedbacks in the students' perception and satisfaction.

A future development of this work can consist in testing the effectiveness of the online course here designed for other university degree courses, to check the benefits also for students of different-topic courses. Furthermore, it could be important to improve the interactive online tutor by developing a new standard plug-in that easily allows the teacher to set the code criteria, to check the quality of the code and allows interoperability between different programming languages.

References

Ambrósio A.P., Moreira Costa F., Almeida L, Franco A., Macedo J. (2011), Identifying Cognitive Abilities to Improve CS1 Outcome, 41st ASEE/IEEE Frontiers in Education Conference, Rapid City, SD, USA.

Amendola D., Miceli C. (2016), Online Physics laboratory for University courses, Journal of E-Learning and Knowledge Society, 12(3), 75-85.

Amendola D., Miceli C. (2018), Online peer assessment to improve students' learning outcomes and soft skills, Italian Journal of Educational Technology, 26(3), 71-84.

Aşkar P., Davenport D. (2009), An investigation of factors related to self-efficacy for Java programming among engineering students, The Turkish Online Journal of Educational Technology (TOJET), 8(1).

Bishop C.M. (2006), Pattern Recognition and Machine Learning, Springer.

Bovo A., Sanchez S., Héguy O., Duthen Y. (2013), Clustering Moodle data as a tool for profiling students, Second International Conference on E-Learning and E-Technologies in Education (ICEEE), 121-126.

Da Re L. (2018), Promoting the academic success: the Formative Tutoring between research and intervention in the experience of the University of Padua, 16(3), 185-199.

De Santis A., Sannicandro K., Bellini C., Minerva T. (2021), Cluster analysis for tailored tutoring system. Q-TIMES WEBMAGAZINE, 3, 265-277.

Di Battista D. (2005), The Immediate Feedback Assessment Technique: A Learner-centered Multiple-choice Response Form, The Canadian Journal of Higher Education, 25(4), 111-131

Epstein M.L., Lazarus A.D., Calvano T.B., Matthews K.A., Hendel R.A., Epstein B.B., Brosvic G.M. (2002), Immediate Feedback Assessment Technique Promotes Learning and Corrects Inaccurate first Responses. Psychol Record 52, 187–201.

Figueiredo J., García-Peñalvo F.J. (2020), Intelligent Tutoring Systems approach to Introductory Programming Courses, Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'20),Association for Computing Machinery, New York, USA, 34–39.

Figueiredo J., García-Peñalvo F.J. (2018), Building Skills in Introductory Programming, TEEM 2018,

García-Peñalvo F.J. (2016), What Computational Thinking Is, Journal of Information Technology Research 9(3).

Gerdes A., Juering J., Heeren B. (2012), An Interactive Functional Programming Tutor, ITiCSE'12, 250-255.

Guelfi M.R., Masoni M., Shtylla J., Formiconi A.R., (2020) Utilizzo di un MOOC in un corso universitario: studio dell'impatto in termini di apprendimento e gradimento, Reports on E-Learning, Media and Education Meetings, 8(1), 166-171.

Gusukuma L., Bart A.C., Kafura D., Ernst J. (2018), Misconception-Driven Feedback: Results from an Experimental Study, Proceedings of the 2018 ACM Conference on International Computing Education Research, 160-168.

Hadjerrouit S. (2008), Towards a Blended Learning Model for Teaching and Learning Computer Programming: A Case Study, Informatics in Education, 7(2), 181-210.

Hackeling G. (2014), Mastering Machine Learning with Scikit-Learn, Packt Publishing.

Hattie J., Timperley H. (2007), The Power of Feedback, Review of educational research, 77(1). 81-112.

Hawi N. (2010), Causal Attributions of Success and Failure Made by Undergraduate Students in an Introductory-Level Computer Programming Course, Computers & Education, 54(4), 1127-1136.

Marwan S., Williams J. J., Price W.T. (2019), An Evaluation of the Impact of Automated Programming Hints on Performance and Learning, Proceedings of the 2019 ACM Conference on International Computing Education Research, 61–70.

Nalli G., Amendola D., Perali A., Mostarda L. (2021), Comparative Analysis of Clustering Algorithms and Moodle Plugin for Creation of Student Heterogeneous Groups in Online University Courses, Applied Sciences, 11(13).

Nalli G, Amendola D, Smith S. (2022), Artificial Intelligence to Improve Learning Outcomes Through Online Collaborative Activities, Proceedings of the 21st European Conference on e-Learning – ECEL 2022, 21(1), 475-479.

Özmen B., Altun A. (2014), Undergraduate Students' Experiences in Programming: Difficulties and Obstacles, 5(3), 9-27.

Richter T., Rudlof S., Adjibadji B., Bernlohr H., Gruninger C., Munz C.D., Stock A., Rohde C., Helmig R. (2012), ViPLab: A Virtual Programming Laboratory for Mathematics and Engineering, Interactive Technology and Smart Education, 9(4), 246-262.

Robins A., Rountree J., Rountree N. (2010), Learning and Teaching Programming: A Review and Discussion, Computer Science Education, 13(2), 137-172.

Shute V.J. (2008), Focus on Formative Feedback, Review of Educational Research, 78(1), 153-189.

Tan J., Guo X., Zheng W., Zhong M. (2014), Case-based teaching using the Laboratory Animal System for learning C/C++ programming, 77, 39-49.

Tan P., Ting C., Ling S. (2009), Learning Difficulties in Programming Courses: Undergraduates' Perspective and Perception, International Conference on Computer Technology and Development, 42-46.

Vellido A., Castro F., Nebot A. (2011), Clustering Educational Data, Handbook of Educational Data Mining, 75-92.

Wing J.M. (2006), Computational thinking, Communications of the ACM, 49(3), 33-35.