



OPEN ACCESS

EDITED BY

Tetiana Vakaliuk,
Zhytomyr State Technological University,
Ukraine

REVIEWED BY

Miha Slapničar,
University of Ljubljana, Slovenia
Etienne Twizeyimana,
University of Rwanda, Rwanda
Euripides Hatzikraniotis,
Aristotle University of Thessaloniki, Greece

*CORRESPONDENCE

Tze Wei Liew
✉ twliew@mmu.edu.my

RECEIVED 30 September 2024

ACCEPTED 11 March 2025

PUBLISHED 21 March 2025

CITATION

Ahmed S, Bukhari SAH, Ahmad A, Rehman O,
Ahmad F, Ahsan K and Liew TW (2025) Inquiry
based learning in software engineering
education: exploring students' multiple
inquiry levels in a programming course.
Front. Educ. 10:1503996.
doi: 10.3389/feduc.2025.1503996

COPYRIGHT

© 2025 Ahmed, Bukhari, Ahmad, Rehman,
Ahmad, Ahsan and Liew. This is an
open-access article distributed under the
terms of the [Creative Commons Attribution
License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or
reproduction in other forums is permitted,
provided the original author(s) and the
copyright owner(s) are credited and that the
original publication in this journal is cited, in
accordance with accepted academic
practice. No use, distribution or reproduction
is permitted which does not comply with
these terms.

Inquiry based learning in software engineering education: exploring students' multiple inquiry levels in a programming course

Sohaib Ahmed¹, Syed Ahmed Hassan Bukhari², Adnan Ahmad³,
Osama Rehman¹, Faizan Ahmad⁴, Kamran Ahsan⁵ and
Tze Wei Liew^{6*}

¹Department of Software Engineering, Bahria University Karachi Campus, Karachi, Pakistan, ²Testing Centre of Excellence, Habib Bank Limited, Karachi, Pakistan, ³Department of Computer Sciences, COMSATS University Islamabad, Lahore, Pakistan, ⁴Cardiff School of Technologies, Cardiff Metropolitan University, Cardiff, United Kingdom, ⁵Faculty of Engineering & Computer Science, Millennium Institute of Technology & Entrepreneurship, Karachi, Pakistan, ⁶Faculty of Business, Multimedia University, Cyberjaya, Malaysia

Learning programming is becoming crucial nowadays for producing students with computational thinking and problem-solving skills in software engineering education. In recent years, educators and researchers are keen to use inquiry-based learning (IBL) as a pedagogical instructional approach for developing students' programming learning skills. IBL instructions can be provided to students through four inquiry levels: *confirmation*, *structured*, *guided*, and *open*. In the literature, most of the IBL applications follow only one form of inquiry level to explore observed phenomena. Therefore, the purpose of this paper is to explore multiple inquiry levels in a programming course taught in a software engineering program. For this purpose, MILOS (Multiple Inquiry Levels Ontology-driven System) was developed using activity-oriented design method (AODM) tools. AODM is an investigative framework to identify the significant elements of the underlying human activity. In MILOS, students were involved in different inquiry levels to answer the questions given related to the programming concepts taught in their classes. Further, MILOS was evaluated with 54 first-year software engineering students through an experiment. For the comparison, Sololearn, an online programming application, was tested with 55 first-year software engineering students. Both these applications were evaluated through *Micro* and *Meso* levels of the M3 evaluation framework. The *Micro*-level investigates usability aspects of MILOS and Sololearn while the *Meso* level explores the learners' performances using these applications. Overall, the results were promising as students outperformed using MILOS as compared to Sololearn application.

KEYWORDS

inquiry based learning, inquiry levels, computer programming skills, software engineering, higher education

1 Introduction

Learning programming enables students' computational thinking, problem-solving and analytical skills in many ways (Martín-Ramos et al., 2018; Vinnervik, 2023). Developing computer programming skills has become one of the important ingredients in software engineering education (Wang and Hwang, 2017). Computer programming literature indicates that such skills pose numerous challenges among students including less motivation and

self-confidence, particularly for novice programmers (Wang and Hwang, 2017; Medeiros et al., 2019). These challenges may be raised due to students' deeper learning of fundamental programming concepts and the inability to apply programming skills to practical solutions (Chao, 2016; Tsai, 2019).

Traditionally, inquiry-based learning (IBL) is established as an effective instructional approach in which learners are involved to investigate the solutions to given problems and creating new knowledge from the underlying phenomena (Liu et al., 2021) and enhanced students' engagement and motivation toward learning domain knowledge (Ahmed et al., 2019; Khan and Ahmed, 2025). Using IBL in programming may improve students' participation for developing design solutions through their inquiry skills (Gunis et al., 2020; Ladachart et al., 2022; Tsai et al., 2022). In recent years, educators and researchers have been interested in developing applications that can assist students in learning programming skills through the IBL instructional approach (Gunis et al., 2020; Yi-Ming Kao and Ruan, 2022).

IBL instructions can be divided into four inquiry levels (Banchi and Bell, 2008): *confirmation*, *structured*, *guided*, and *open*. Confirmation inquiry is a teacher-led instruction where the description of a problem, procedure, and solution is given to students to reinforce an already introduced topic. Structured inquiry is the second level in the continuum of the levels of inquiry (Tan et al., 2018), where students investigate the unknown phenomena through known problems and procedures. At the third level, Guided inquiry possesses some challenges for students in investigating phenomena through their procedures. At the highest level, in Open inquiry, students need to identify an inquiry problem using their procedures for the underlying observed phenomena.

In literature, IBL instructions are prevalent in science education and computer programming with a single level of inquiry (Tsai, 2019; Gunis et al., 2020; Rahmat and Chanunan, 2018; Umer et al., 2017). Conversely, there are a few proponents in literature that highlight the use of multiple inquiry levels in science education (Chaudhary and Lam, 2021; Muskita et al., 2020) and computer programming (Deshmukh and Raisinghani, 2018). According to these studies, multiple inquiry levels can provide better understanding of the domain knowledge and encourage students to learn observed phenomena having different challenges and perspectives (Chaudhary and Lam, 2021; Muskita et al., 2020; Deshmukh and Raisinghani, 2018). However, limited studies focus on the evaluation of multiple inquiry levels in software engineering education, specifically for computer programming skills (Deshmukh and Raisinghani, 2018). Therefore, this allows us to explore multiple inquiry levels in a computer programming course taught in software engineering program, which was not explored earlier in the literature.

On the other hand, ontology-driven approaches have been used in literature for conducting science inquiry investigations. For instance, in ThinknLearn (Ahmed and Parsons, 2013), ontology-driven approach used as an instructional design model and give adaptive hints (guidance) to learners while conducting science laboratory experiments. In a similar vein, Lin and Lin (2017) implemented an ontology-driven approach for providing contextual meaning of museum exhibits and virtual user profiling of museum visitors. In both these studies, domain ontologies are used for implementing ontology-driven approach. Hence, we have developed a system namely MILOS (Multiple Inquiry Levels Ontology-Driven

System) using inquiry domain ontology. Further, AODM (Activity Oriented Design Method) tools (Ahmed et al., 2012a) are used for designing MILOS by which university students can have an interactive environment for exploring multiple inquiry levels to learn computer programming skills. For evaluation purposes, Micro and Meso levels of the M3 evaluation framework (Vavoula et al., 2009) have been used. The M3 evaluation framework, AODM tools, results and analyses are discussed later in this paper.

2 Current state of the art

IBL, an effective pedagogical approach where learners are required to solve the given problems through experimentation and observed phenomena (Liu et al., 2021). In recent literature, this instructional approach has been prevalent in developing applications that can guide learners to enhance their programming skills (Gunis et al., 2020; Yi-Ming Kao and Ruan, 2022). Such IBL instructions can be defined into four inquiry levels (Banchi and Bell, 2008): *confirmation*, *structured*, *guided*, and *open*. In literature related to education, a single inquiry level is mostly used by students for various purposes. In recent times, it has been evident that researchers and educators are emphasizing the use of multiple inquiry levels in science education (Chaudhary and Lam, 2021; Muskita et al., 2020). However, there are very limited empirical studies that focus on multiple inquiry levels for learning computer programming skills. The current state of the art in the use of inquiry levels in empirical studies for various domains including science and programming is depicted in Table 1.

In most of the literature related to IBL, a single level of inquiry has been explored. In one of these studies, an open inquiry was conducted on the metacognitive skills of senior high school students (Rahmat and Chanunan, 2018). This research concluded that an open inquiry significantly improved both types of students' metacognitive skills; low and high academic abilities while performing biological science learning activities. In another research, Kallas and Pedaste (2018) developed an android-based mobile inquiry learning application in which students were involved in planning 8th Grade Chemistry experiments through guided inquiry. This application helps students to choose the appropriate laboratory equipment and substances through feedback. It further guides the students in determining the experimental steps.

In a similar vein, MAPILS (Umer et al., 2017) and ThinknLearn (Ahmed and Parsons, 2013) were designed in which guided inquiry was explored. In MAPILS (Umer et al., 2017), 8th-grade science students were involved to learn the concepts of the plants, Palm tree, Lotus, Pine tree, and Mushroom. This augmented learning application assists students to focus on a particular plant through their mobile devices and then perform inquiry learning activities. ThinknLearn (Ahmed and Parsons, 2013) was another guided inquiry learning application designed for high school students. In this mobile application, students were guided to generate hypotheses on a science topic "heat energy transfer" by using their critical thinking and inquiry skills to describe the underlying phenomena.

weSPOT (Mikroyannidis et al., 2013) was another initiative for guided IBL activities in secondary and higher education. In this research project, two scenarios of different domains including microclimates, and the energy efficiency of buildings were used. In this project, students enhanced their inquiry and reasoning skills through animated virtual

TABLE 1 Inquiry levels conducted in programming and other applications.

Applications/authors	Inquiry domain	Inquiry level	Learners
Rahmat and Chanunan (2018)	Science	Open	Senior high school students
Kallas and Pedaste (2018)	Science	Guided	Secondary school students
MAPILS (Umer et al., 2017)	Science	Guided	High school students
ThinknLearn (Ahmed and Parsons, 2013)	Science	Guided	High school students
weSPOT (Mikroyannidis et al., 2013)	Science	Guided	Secondary school students and teachers
EduVenture (Jong and Tsai, 2016)	Social Science	Guided	Secondary school students and teachers
Chaudhary and Lam (2021)	Science	Structured, guided, and open	University students
Muskita et al. (2020)	Science	Structured, guided, and open	University students
Tsai (2019)	Programming	Guided	University students
Gunis et al. (2020)	Programming	Structured	Secondary school students
SoloLearn (Deshmukh and Raisinghani, 2018)	Programming	Guided and structured	University students
jeKnowledge (Martín-Ramos et al., 2018)	Programming	Guided	Senior high school students and first-year university students
Lab4CE (Venant et al., 2017)	Programming	Guided	University students

models. In the second scenario “energy efficiency of buildings,” teachers were also involved in answering IBL activities. Similarly, EduVenture (Jong and Tsai, 2016) was a research project that targets social science inquiry learning. In this project, LOCALE (Location-Oriented Context-Aware Learning Environment) was an electronic resource composed by teachers before students’ field trips. EV-eXplorer was a mobile application designed for students who were performing inquiry activities during field trips. This research was evaluated by secondary school students and liberal studies teachers from 18 schools and indicated that participants gain knowledge through guided inquiry activities.

There are few studies in literature that follow multiple levels of inquiry in science education. In one of the studies (Chaudhary and Lam, 2021) in which lab-based modules were designed for first-year biomedical engineering students to evaluate multiple inquiry levels including structured, guided, and open inquiry. In these modules, the first two modules were based on structured inquiry in which students were assigned to analyze experimental data through known procedures. The other two modules were designed on guided inquiry principles in which students were involved to analyze experimental data through faulty laboratory procedures. In these modules, students were expected to deviate from the expected outcomes and proposed reasons for such deviations. The final module was based on a design project where students were involved in an open inquiry activity.

Another research was conducted with the University students to perform multiple inquiry levels; structured, guided, and open in the domains of plant Morphology, Ecology, and Physiology (Muskita et al., 2020). In this research, these multiple inquiry levels were investigated through worksheet design of learning activities; problem definition, hypothesis generation, data collection, analysis, and conclusion to enhance critical thinking and creative learning skills. This research was designed in an environment where students were involved in multiple inquiry-level learning activities.

In the literature related to programming-based inquiry learning, many studies have been found where a single level of inquiry was used. For instance, a study was designed for administrating basic computer programming tests among the students from the general education program of the University (Tsai, 2019). This research used AI2 (App Inventor 2), a web-based environment for designing

Android applications in a design-based learning (a form of inquiry-based learning) environment. The research explored the effectiveness of such a learning environment in improving students’ programming skills. jeKnowledge (Martín-Ramos et al., 2018) was developed to engage high-school and first-year undergraduate students to learn introductory Arduino programming. This application guides students through hands-on projects using a peer-coaching strategy. Similarly, Lab4CE (Laboratory for Computer Education) (Venant et al., 2017), a web-based virtual learning environment was developed for students to assess physical resources from anywhere and enhance their inquiry skills. Students learn Shell programming concepts through their virtual machines in this virtual laboratory.

In another research, inquiry-based Python programming was implemented at secondary schools (Gunis et al., 2020). The research demonstrates that such structured inquiry played a vital role in developing students’ cognitive learning and problem-solving skills while learning Python programming language. On the other hand, Sololearn C++ (Deshmukh and Raisinghani, 2018) was used by which students can learn object-oriented programming (OOP) concepts through multiple inquiry levels; guided and structured. In this research, the collaborative IBL approach was applied to enhance students’ programming skills by reinforcing concepts taught in their classrooms.

The current state of the art highlighted that a single inquiry level has been used in most of the studies. However, few studies focus on multiple inquiry levels in science education (Chaudhary and Lam, 2021; Muskita et al., 2020). In contrast, only a single study has been found where multiple inquiry levels are explored for learning computer programming skills (Deshmukh and Raisinghani, 2018). This allows us to explore such learning environments in this research where multiple inquiry levels are evaluated in a programming course of software engineering education.

3 Materials and methods

Activity Oriented Design Method (AODM) is an investigative framework for analyzing and characterizing technology-enhanced learning practices and follows basic principles of activity theory (Mwanza-Simwami, 2011). This method has been used in the

TABLE 2 Eight step model (adapted from Ahmed et al., 2012a).

Step	Activity	Questions	Answers
1	Activity of interest	What sort of activity am I interested in?	Environment having multiple inquiry levels
2	Objective	Why is the activity taking place?	Performing multiple inquiries
3	Subjects	Who is involved in carrying out the activity?	University 1st year students
4	Tools	By what means are the subjects performing this activity?	MILOS/Sololearn
5	Control	Are there cultural norms, rules, or regulations governing the performance of the activity?	University ethics (curriculum)
6	Communication	Who is responsible for what and how roles are organized?	Mode of interaction and support of co-workers (teacher/student/researcher)
7	Community/context	What is the environment in which this activity is carried out?	Programming-based inquiry learning environments
8	Outcome	What is the desired outcome of carrying out this activity?	Reinforced knowledge of programming concepts

literature for technology-enhanced learning and design (Ahmed et al., 2012b; Mwanza-Simwami, 2017; Pertiwi, 2018). AODM has four methodological tools that assist in identifying the significant elements of any underlying human activity. These tools include (i) eight-step model (ii) activity notation (iii) technique for generating research questions and (iv) technique for mapping operational processes. In this research, these methodological tools are applied to generate research questions and identify an operational process. For the evaluation of MILOS, two levels of the M3 evaluation framework (Vavoula et al., 2009) were used; *Micro* and *Meso*. The *Micro*-level examines usability aspects of MILOS while the *Meso* level investigates the learners' experiences during multiple inquiry level activities.

3.1 AODM tools

3.1.1 AODM tool 1: eight step model

The Eight-Step model describes the questions at each step of an activity system of the underlying problem. This model has been adopted in this research by adding the "Answers" column to answer the question asked at each step of the activities performed during this research as shown in Table 2.

3.1.2 AODM tool 2: activity notation

AODM's activity notation was designed to decrease complexity by decomposing the activity system into sub-activity triangles (Mwanza-Simwami, 2011). The sub-activity triangles are united through the shared object of the main activity system. This helps researchers to perform a detailed analysis of sub-activity triangles for generating research questions about an underlying human activity (Ahmed et al., 2012a).

This research does not focus on the issues related to communication and control while producing sub-activity triangles. Thus, for generating sub-activity triangles, the other four remaining components can be analyzed (Subject, Tool, Object, Context): (i) (Subject, Tool, Object); (ii) (Subject, Tool, Context); (iii) (Subject, Object, Context); and (iv) (Tool, Object, Context). In contrast, this research targets the proposed tool (MILOS) and objective (performing multiple inquiries). Therefore, two sub-activity triangles are being used (Subject, Tool, Object) and (Tool, Object, Context) as shown in Figure 1.

3.1.3 AODM tool 3: generating research questions

This technique generates research questions to facilitate the detailed concept modeling of the data gathering and analysis from an activity theory perspective and helps to identify relationships and contradictions within the activity system (Mwanza-Simwami, 2011). Two research questions are derived from the "Tool" and "Object" components of the activity system as depicted in Figure 1. These research questions can be used to analyze the applicability of a designed tool "MILOS" for performing multiple inquiry levels in a programming course of software engineering education.

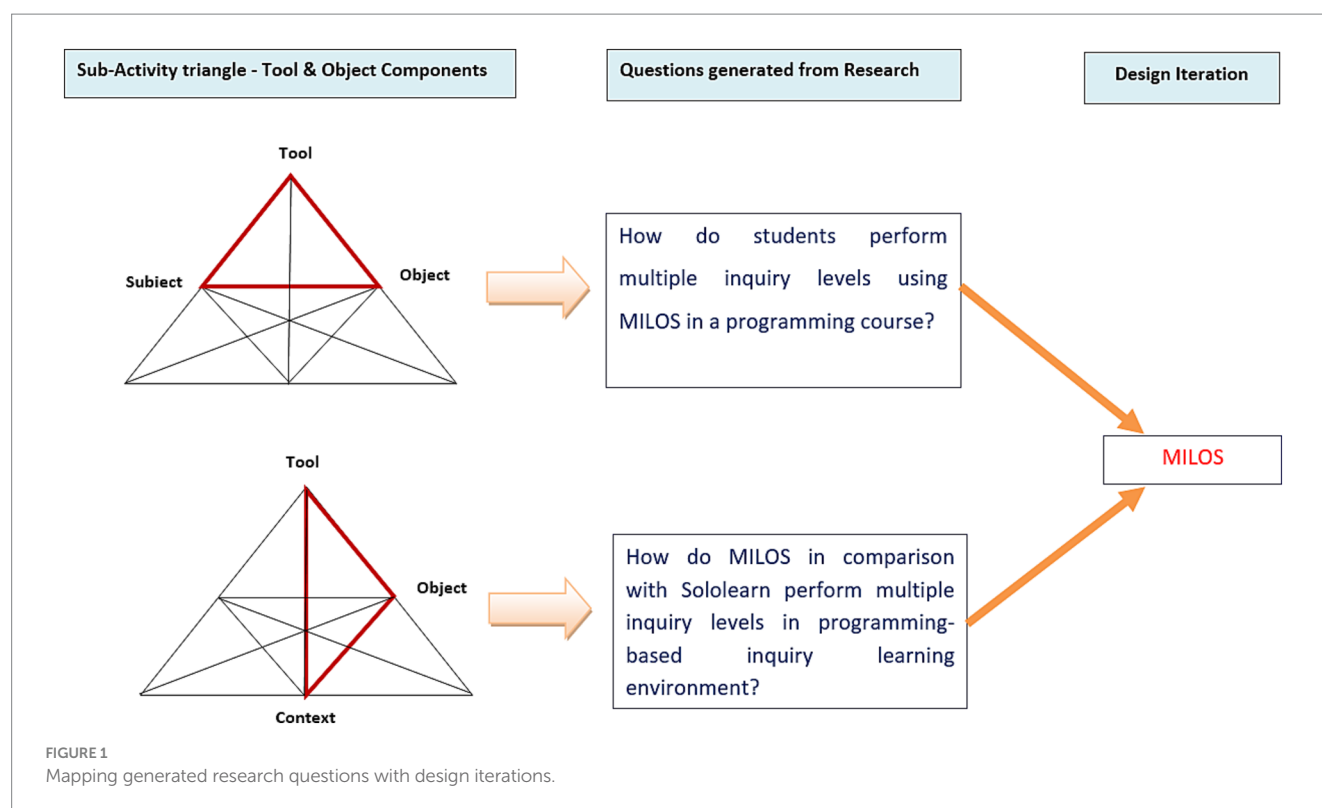
RQ1: How do students perform multiple inquiry levels using MILOS in a programming course?

RQ2: How do MILOS in comparison with Sololearn perform multiple inquiry levels in programming-based inquiry learning environment?

3.1.4 AODM tool 4: mapping operational processes

This technique helps to understand research findings by providing a graphical representation of the activities that take place in an underlying domain (Mwanza-Simwami, 2011). Using this tool, generated research questions are mapped into design iterations within the activity system and may produce design insights for further refinement (Ahmed et al., 2012a).

In the first sub-activity triangle subject-tool-object, a research question is generated about the use of a designed tool "MILOS" for performing multiple inquiry levels in a programming course. For answering this research question, an inquiry ontology was developed and used for generating multiple-choice questions and distractors (answers) for performing multiple inquiry level tasks in a designed tool "MILOS" (see Section 4 for details). In the second sub-activity triangle tool-context-object, a research question is generated regarding the use of MILOS for performing multiple inquiry levels in a programming-based inquiry learning environment in comparison with Sololearn, an online programming environment. For this purpose, MILOS and Sololearn were tested with 1st year engineering students enrolled in a programming course. Both research questions are answered through a design iteration in which an application "MILOS" was designed and tested with engineering students. This research question is answered through micro



and meso levels of the M3 evaluation framework as mentioned in Section 3.2.

3.2 Evaluation method

For the evaluation purposes of the designed system “MILOS,” the two levels including Micro and Meso of the M-3 evaluation framework were applied (Ahmed and Parsons, 2013). The Micro-level evaluation identifies the utility of applications in terms of guidance to students while performing inquiry tasks. Three usability aspects (Chua et al., 2004); *learnability, operability, and understandability* mentioned in Ahmed and Parsons (2013) were evaluated at the Micro level. For the Meso level evaluation, students’ performances were investigated. Both levels are used to answer research question 2 (RQ2) of this research study.

According to the literature related to computer programming, Sololearn is the only application where multiple inquiry levels including structured and guided are used for learning programming skills (Deshmukh and Raisinghani, 2018). Thus, MILOS is compared with Sololearn at Micro and Meso level evaluations as depicted in Figure 2.

3.2.1 Experimental design

In the experiment, students are divided into two groups experimental and control groups. The experimental group was asked to do multiple inquiry level activities with this process: “Get multiple inquiry level questions → Get hints about inquiry questions → Answer inquiry questions about the given topic → Calculate learning performance marks → Evaluate usability and quality aspects.” The compared group was examined through the Sololearn application. In this approach, students need to answer the asked inquiry questions related to multiple inquiry levels; structured and guided. The control group follows the sequence of

activities: “Get multiple inquiry level questions → Get hint about inquiry questions → Answer inquiry questions → Calculate learning performance marks → Evaluate usability and quality aspects.” Both these groups have received the identical inquiry tasks and evaluate knowledge about the topic “Variables.” Learning about programming concepts using MILOS and Sololearn are considered the independent variables of the experiment, while the technological usability and learning performance (scores) are the dependent variables. The evaluation of learning performance determines how well each student has learned about the given topic and performs multiple inquiry levels.

For micro-level evaluation, students were asked to answer a series of questions on a five-point Likert scale regarding usability and quality aspects of MILOS and Sololearn. These questions are described in Table 3. For Meso-level evaluation, marks obtained by students of each group are calculated on the inquiry questions asked during the experiment. In MILOS, 10 inquiry questions (4 each of structured and guided inquiries while 2 open inquiry questions) were asked while in Sololearn, 8 inquiry questions (4 each of structured and guided inquiries) were answered. The results and analyses are discussed in the following section.

3.2.2 Participants

The experiment was conducted in the first 2 weeks of October 2023 by the first-year students of the Department of software engineering at Bahria University Karachi Campus, Pakistan. One hundred and nine students aged between 18 and 20 years voluntarily participated, divided into two groups: Experimental and Control groups. In the experimental group, 62 were male while remaining 47 were female participants. This group uses MILOS application. The control group investigated multiple inquiries through Sololearn in the first week in which 32 males and 23 females participated. The

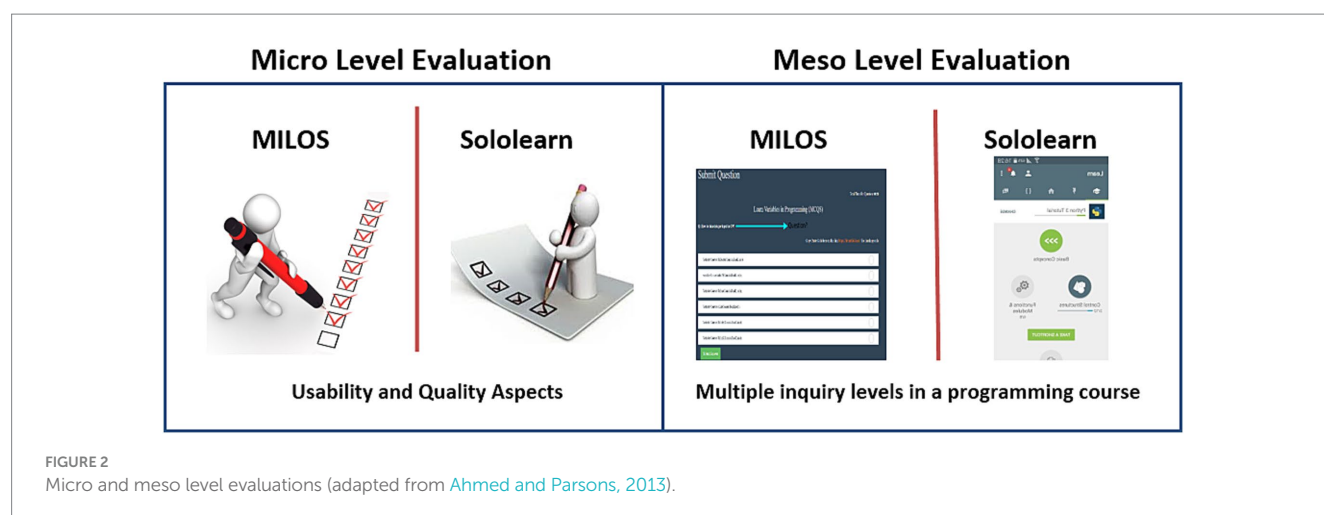


TABLE 3 Usability and quality sub-characteristics along with descriptions.

Characteristic	Sub-characteristics	Description
Usability aspects	Learnability	Can a student learn to use an application easily?
	Operability	Can a student use the application without much effort?
	Understandability	Does a student comprehend how to use the application?
Quality aspects	Learning content	Does a student feel that the learning component is of high quality?
	Metaphor	Does a student have an overall vision of the inquiry learning process?
	Interactivity	Is a student able to interact with a given application?

experimental group was evaluated in the second week of the experiment where 30 male and 24 female students participated.

Initially, the purpose of the research activity was explained to all participants and invited them to fill in individual consent forms. These forms have given instructions regarding the conducting experiment. In this research activity, 55 participants were asked to examine technological usability and inquiry learning through Sololearn. On the other hand, the experimental group was comprised of 54 students from both departments. At the time of the experiment, all students had already covered the concept of variables in their theoretical classes. However, they did not conduct any prior lab work on the given topic.

4 MILOS: a multiple inquiry levels ontology-driven system

In consultation with programming course teachers, we have chosen the topic “Variables” from the course “Object-Oriented Programming.” For this purpose, we have designed a system called MILOS (A multiple inquiry levels ontology-driven system) by using ADOM tools as mentioned in Figure 1. The technical architecture is divided into three main components: client, web server, and ontology. This architecture is adapted from Ahmed et al. (2012a) as depicted in Figure 3.

4.1 Client component

This client component is designed to demonstrate multiple inquiry levels in a programming course. Users/clients can

be connected to the web browser through web and mobile. The client component can access the ASP. Net MVC web server using. Net Framework 4.5. Further, the IIS (Internet Information Services) server is used in the designed system, MILOS. IIS server runs on Windows systems that accept requests from the client component and return the requested web pages to the client component.

Students need to register in the designed system, MILOS, for performing multiple inquiry levels. The system will give inquiry tasks in the form of questions. These questions have been posed to students related to the concept “Variables” of a programming course. In this inquiry, students are provided with suitable methods so that they can apply to answer given inquiry questions as depicted in Figure 4. For this phase, they define results on their own. Relevant suggestions are also provided for students for performing this inquiry level.

For a guided inquiry, students are free to design their methods and describe their results. For this scenario, an inquiry question is given to a student for understanding and exploration of a given phenomenon. However, for the other three phases, students are required to use their own methods and describe the results of the given inquiry questions as shown in Figure 5. The guidance is also provided to students through relevant suggestions during these inquiry activities.

In an open inquiry, students are free to explore their problems by designing their methods and then mentioning their outcomes. For this scenario, an inquiry programming question is given to students for conducting an open inquiry as depicted in Figure 6. In this inquiry, no guidance is provided to students through any means. They are

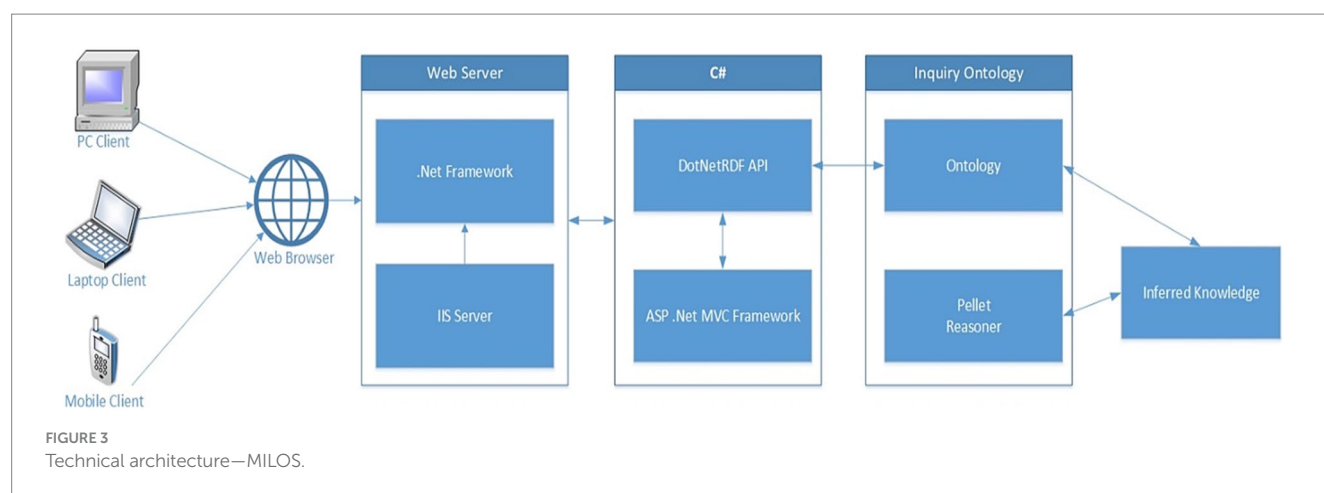


FIGURE 4
Structured inquiry.

invited to write a programming code and submit it to a teacher for further evaluation.

4.2 Web server component

This component provides a connection between client and ontology components. Users/clients send a request to an IIS server (webserver) component using the .Net framework to retrieve relevant information about the underlying concept “Variables” in an object-oriented programming course. This component connects with the ontology component for extracting content from the inquiry ontology. For this purpose, DotNetRDF¹ API and ASP .Net

MVC framework are used for retrieving information from ontology. DotNetRDF is a flexible API written in C# and can be stored in RDF/OWL format.

4.3 Inquiry ontology component

This component describes an inquiry ontology that consists of information regarding variables in a programming language. This inquiry ontology is defined by six main concepts; inquiry level, instructional model, function, variable name, variable type, and variable as depicted in Figure 7.

The concept “Inquiry_Level” defines all four inquiry levels: Confirmation, Structured, Guided, and Open. In the concept “Instructional_Model,” five types are defined as mentioned by Bybee (2019). These types are engagement, exploration, explanation, elaboration, and evaluation. For instance, in the guided inquiry level,

¹ <https://dotnetrdf.org/>

Submit Question

Total Time for Question: 00:28

Learn Variables in Programming (MCQS)

Q: How to take integer input in C#? → Question?

Copy Paste Code here on this link <https://dotnetfiddle.net/> for checking code

- ☐ Variable=Convert.ToDouble(Console.ReadLine());
- ☐ variable=Convert.ToInt32(Console.ReadLine());
- ☐ Variable=Convert.ToFloat(Console.ReadLine());
- ☐ Variable=Convert.ToInt(Console.ReadLine());
- ☐ Variable=Convert.ToInt64(Console.ReadLine());
- ☐ Variable=Convert.ToInt32(Console.ReadLine());

Submit Answer

FIGURE 5
Guided inquiry.

Learn Variables in Programming

Q: Write a program to take two input from user to show name and age?

Submit Answer

FIGURE 6
Open inquiry.

two types of instructional models (engagement and exploration) are teacher-led while others are student-initiated. For this purpose, the ontology presents function (method) as depicted in Figure 7.

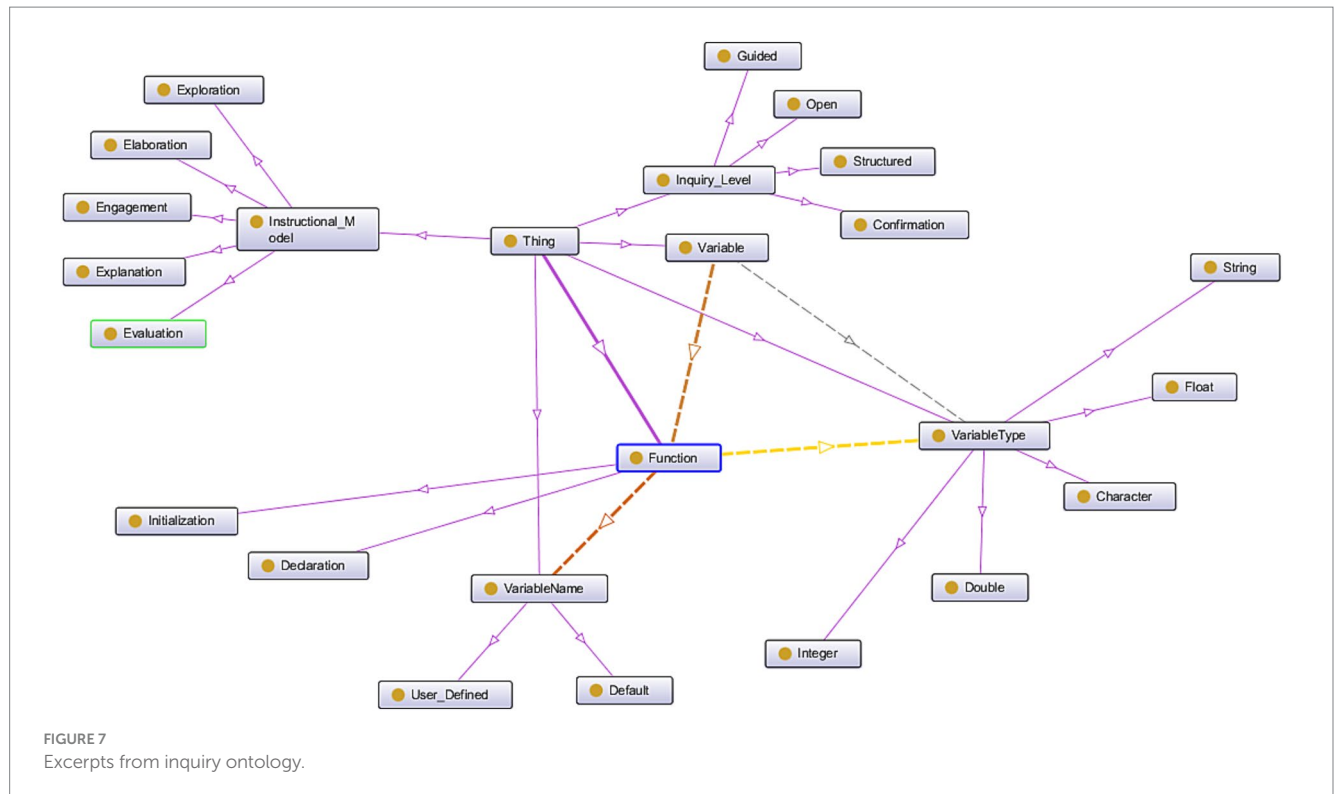
The other concepts related to the topic “Variables” are defined in the ontology as learning concepts. The concept “VariableType” is categorized into five types that record the variable type of each variable defined in the given inquiry question. Another concept of “VariableName” is defined in the ontology that covers the description regarding default and user-defined variable names. Each variable name can have any variable type. Further, the concept “Function” is also mentioned in the ontology. This concept defines the declaration and initialization of any variable. For instance, the declaration sub-concept defines the variable with the variable name and variable type. On the other hand, the initialization sub-concept describes the variable with a variable name, variable type, and value.

Multiple choice questions (MCQs) are one of the prevalent assessment methods in traditional and electronic learning

environments for assessing students’ knowledge about a specific topic (Papasalouros and Chatzigiannakou, 2018). The ontology-driven approach can be used for generating MCQs and answers (correct or distractors) through the concepts defined in the inquiry ontology. In this system, inquiry questions (MCQs) are generated through the inquiry ontology for evaluating students’ knowledge regarding variables in object-oriented programming. Further, the answers (correct and distractors) to the questions are randomly generated. Hermit reasoner² is used for inferring knowledge from the inquiry ontology. Moreover, SPARQL,³ a query language for RDF (Resource Description Framework) is used for retrieving relevant information for generating inquiry questions and hints.

² <http://www.hermit-reasoner.com/>

³ <https://www.w3.org/TR/rdf-sparql-query/>



5 Results and analyses

This research study used Micro and Meso levels of the M3 evaluation framework for answering research question 2 (RQ2). The purpose of this evaluation is to compare the usability and quality aspects of MILOS and Sololearn and measure the learning performances of the participants using MILOS and Sololearn. The corresponding results are discussed below.

5.1 Micro level evaluation

In the Micro-level evaluation, the technological usability of both applications MILOS and Sololearn were evaluated and compared through a questionnaire. The questionnaire was asked for by the participants of MILOS and Sololearn regarding their overall inquiry learning experiences of these applications in a programming course. The questionnaire consists of 9 questions including an open-ended text box for additional remarks from the participants. These questions were based on usability aspects, usability (*learnability, operability, understandability,*) and mobile quality (*learning content, metaphor, interactivity*) as described in Table 4. A five-point Likert scale was used; 5 was “strongly agree” and 1 was “strongly disagree.” The responses from the participants showed that the questionnaire was reliable (the value of Cronbach alpha is 0.89).

For the usability aspects, the first four questions were answered by the participants. Q1 was asked about the *learnability* of both applications; MILOS and Sololearn. The responses to question “Q1” indicated that learnability was high in both applications. However, in comparison, participants were more satisfied with MILOS. One of

them described his satisfaction as “... it is very easy to use.” In another example, a student mentioned that “...easy application to learn about variables.” These results were also analyzed through one-sample *t*-test ($t_{107} = 2.81, p < 0.01$ for Q1).

The other usability aspect (*operability*) was asked through question “Q2” as mentioned in Table 4. According to the results, the ratings on “Q2” were promising. In both applications, participants have not given enough effort to complete the inquiry tasks ($t_{107} = 2.35, p < 0.05$ for Q2). In comparison, participants are slightly more satisfied with MILOS as compared to Sololearn. A couple of the students informed about MILOS as “...not much effort is required for using this application” and “nice application for learning variables with much less effort required.” Regarding Sololearn, one of the students commented “it is a very easy and good application, but I believe some challenging questions may be given.” The Sololearn application follows structured and guided inquiry levels therefore, it initially describes the method and solution and then, asks MCQ related to that topic. The challenging part may come with an open inquiry. However, in MILOS, structured, guided and open inquiries were applied. Therefore, participants are more satisfied with MILOS in comparison with Sololearn.

The sub-characteristic (*understandability*) of these applications were tested by asking two questions “Q3” and “Q4.” The responses in Q3 and Q4 revealed that participants were satisfied with both applications. However, MILOS’s participants showed promising results as compared to Sololearn. One sample *t*-test verified these explanations ($t_{107} = 2.25, p < 0.05$ for Q3; $t_{107} = 4.47, p < 0.01$ for Q4). Most of the participants were happy with the guidance provided by MILOS. One of the positive responses was “The guidance provided by this application is remarkable as the suggested hints help me to understand variables.” However, some participants especially female students were not satisfied with the understanding of how to use

TABLE 4 Usability and quality aspects results of MILOS and Sololearn.

No.	Questions	Usability sub-characteristics	Mean $\pm$ standard error (MILOS)	Mean $\pm$ standard error (Sololearn)
Q1	Is this application easy to use?	Learnability	4.22 $\pm$ 0.13	3.67 $\pm$ 0.14
Q2	Do you need much effort to complete the given inquiry tasks?	Operability	4.24 $\pm$ 0.12	3.78 $\pm$ 0.14
Q3	Do you easily understand how to use the given application?	Understandability	4.01 $\pm$ 0.14	3.56 $\pm$ 0.13
Q4	Do you think this application provides enough guidance to learn about the given topic?	Understandability	4.24 $\pm$ 0.12	3.36 $\pm$ 0.15
Q5	Does this application increase your curiosity to inquire and learn about the given topic?	Learning content	4.14 $\pm$ 0.13	3.16 $\pm$ 0.18
Q6	Does this application give relevant suggestions to the given inquiry questions?	Learning content	3.94 $\pm$ 0.10	3.54 $\pm$ 0.15
Q7	Does this application help you to know an overall inquiry learning process?	Metaphor	3.79 $\pm$ 0.11	3.34 $\pm$ 0.16
Q8	Is it easy to find out the information about the given topic?	Interactivity	4.01 $\pm$ 0.11	3.58 $\pm$ 0.16
Q9	Is it an overall interactive inquiry learning application?	Interactivity	3.83 $\pm$ 0.11	3.47 $\pm$ 0.13

MILOS. One of the participants described “I was confused while using this application because some questions have a description about the methods, and some have missed such information.” This is because MILOS was designed with the purpose of giving inquiry tasks based on multiple inquiry levels. Therefore, few participants were dissatisfied to some extent with learning the topic through multiple levels. For Sololearn, participants felt displeasure regarding the guidance provided by the application. In one instance, a female student stated as “it just gives answers straightaway instead of giving hints for thinking about the solution.” In another instance, a male student described his discomfort as “this application did not challenge me, it would be better if it provides some challenges to us so that we can be more excited to learn from it.” Overall, it has been observed that the MILOS participants were more satisfied in comparison with Sololearn participants.

The mobile quality aspects of MILOS and Sololearn were examined through five questions (Q5–Q9) as described in Table 4. Q5 and Q6 are focused on *Learning content*, one of the mobile quality aspects. Overall results showed that the participants were satisfied with MILOS as compared to Sololearn. For question 5, one of the participants said, “I was so curious to learn computer programming using this application (MILOS).” In another instance, a participant mentioned “Application (MILOS) increases my curiosity to learn about variables in programming.” On the other hand, participants showed some reservations about Sololearn. They expressed their views as “A good application to learn about programming, however, it does not increase my curiosity because the answers given in this application were straightforward.” And “this application (Sololearn) did not pose challenging questions. Therefore, it loses my interest while learning.” For question 6, the responses on both applications MILOS and Sololearn were positive. However, MILOS received better feedback than Sololearn. One of the participants responded as “these suggestions given in the application (MILOS) were relevant.” One of the participants indicated that “these hints are relevant to the given topic, however, the options (answers) were very close to each other in the application (MILOS). This makes us more challenging to give answers.” These questions and options (answers) including distractors were randomly generated from the inquiry ontology concepts. In a similar vein, Sololearn participants were satisfied with the relevant suggestions on the given topic. For instance, a participant mentioned “Sololearn is a

good application for learning to code. The suggestions given are a complete description. Anyways, I enjoyed it.” One participant showed her discomfort as “this (Sololearn) has not provided any suggestions. In fact, it gave me a complete description of a given question. I think the suggestions should have been given while I was answering the questions.” This indicates that Sololearn has not provided any such guidance as compared to MILOS. The result of a one-sample *t*-test confirmed these findings ($t_{107} = 4.32$, $p < 0.01$ for Q5; $t_{107} = 2.13$, $p < 0.05$ for Q6).

Question 7 covers one of the mobile quality aspects (i.e., Metaphor). The responses to Question 7 showed that most of the participants were marginally satisfied while using both applications; MILOS and Sololearn. However, the participants of MILOS were given slightly better responses than Sololearn participants. The result of a one-sample *t*-test confirmed these findings ($t_{107} = 0.91$, $p < 0.05$ for Q7). One of the MILOS participants indicated “it is a good application, but I do not have an idea about what type of inquiry process I am involved in.” Other participants mentioned that “different types of questions were posed during this learning process. Some of the questions provided guidance but some of them were very straightforward.” These comments indicated that some participants did not understand the rationale behind the application (MILOS). The application was designed to provide learning through three different inquiry levels: structured, guided, and open. However, this may be further improved for a better understanding of participants in their overall inquiry learning process. On the other hand, Sololearn participants responded, “I am not aware of the inquiry learning process while using this application (Sololearn).” In another instance, the participant mentioned “it provides me a direct answer, I have no idea how my inquiry learning process is going.” This indicates that Sololearn is not providing much help in understanding the overall inquiry learning process.

In responses to Q8 and Q9, the results of other mobile quality sub-characteristic, *interactivity* showed that the MILOS is considerably an interactive application as compared to Sololearn ($t_{107} = 2.12$, $p = 0.05$ for Q8; $t_{107} = 2.04$, $p < 0.05$ for Q9). For question 8, the MILOS participants were happy about the information received about a given topic. The comments were as “... very easy to find relevant information about the variables” and “it is a nice application that helps me to find information about the topic.” On the other hand, the Sololearn participants were marginally satisfied with the information

received about the topic given. One of the responses was, “...nice application but the information was very straightforward, not very much interactive.” The other participant indicated that “this (Sololearn) can be more interactive in terms of guiding us.” However, there are few positive responses about Sololearn. The responses are, “a very interactive application in which I can see other members’ comments on each topic. It helps me to learn about a given topic” and “a very good learning tool for programming.” Overall, the responses of both applications are satisfactory, however, MILOS has received better responses as compared to Sololearn.

For MILOS, the responses to Q9 were straightforward as most of the students highlighted that MILOS is an interactive application. One response was “a very interactive programming learning application that guides me to learn about variables.” In another instance, one of the respondents said, “a very interactive application to learn about programming.” Some of the participants were not satisfied with the interactivity of the application MILOS. One of them mentioned is, “a good application, however, it can be designed as a game, or some virtual characters can be added to make it more interactive.” This comment showed that MILOS may be further improved in terms of interactivity. In comparison with Sololearn, respondents were satisfied with the interactivity of the application. One of the respondents said, “...interactive application as it displays comments of other learners on the same topic.” “It is an interactive application to learn about programming. However, no suggestions are displayed in this application. Instead, a straightforward answer showed up at the end.” Another comment about Sololearn was, “a good learning application but it may be further improved.” The overall results of Q9 for both applications were marginally significant. This shows that students faced some challenges

in terms of interactivity and the use of these applications in learning about the given programming tasks.

In summary, the overall questionnaire responses to both applications were significant. In comparison, the responses of MILOS were more encouraging than Sololearn. This indicates that MILOS has considerably better usability and quality aspects than Sololearn. However, some new approaches may be implemented to make these applications more interactive.

5.2 Meso level evaluation

At this evaluation level, the learning performances of the students of both comparative groups; Experimental (MILOS) and Control (Sololearn) were investigated. The activity consists of multiple inquiry levels; structured, guided, and open. All three levels were evaluated via MILOS application, however, due to limitations in the Sololearn application, only two levels including structured and guided inquiry levels were explored as mentioned in Figures 8, 9. MILOS application asks ten random inquiry questions (4 questions of structured and guided inquiry each while 2 questions of open inquiry were asked) as shown in Figures 4–6. However, in the Sololearn application, eight inquiry questions were asked comprising structured and guided inquiry types. In MILOS, two hints related to each asked inquiry question were given. The application further maintained “3” marks if a student submits the correct answer to each asked inquiry question. If a student uses the first hint, then he will secure “2” marks. However, if he wants further assistance, then the second hint will be provided but he will get a “1” mark only for his corrected answer. And if he is

Variables

Examples of variable declarations:

```
class MyClass {
    public static void main(String[] args) {
        String name = "David";
        int age = 42;
        double score = 15.9;
        char group = 'Z';
    }
}
```

JAVA

Try it Yourself

char stands for character and holds a single character.

Another type is the **Boolean** type, which has only two possible values: **true** and **false**.

This data type is used for simple flags that track true/false conditions.

For example:

```
boolean online = true;
```

JAVA

You can use a comma-separated list to declare more than one variable of the specified type. Example: `int a = 12, b = 11;`

Back

Continue

Variables

Drag and drop from the options below to have a valid Java program.

```
class Apples {
    public static void main(String[] args) {
         name = "John";
         age = 24;
         height = 189.87;
    }
}
```

String

void

int

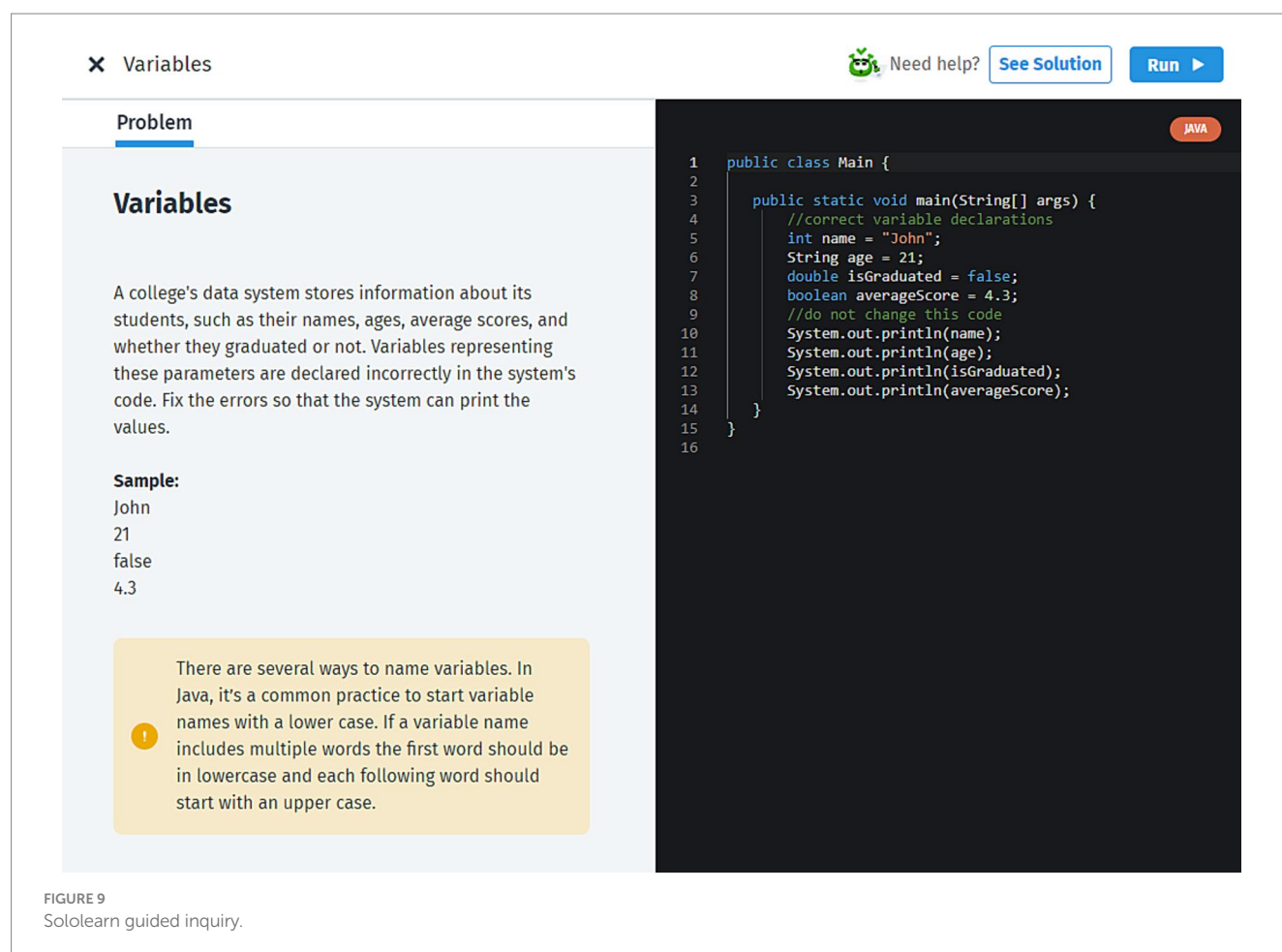
double

class

true

FIGURE 8

Sololearn structured inquiry.



Variables

A college's data system stores information about its students, such as their names, ages, average scores, and whether they graduated or not. Variables representing these parameters are declared incorrectly in the system's code. Fix the errors so that the system can print the values.

Sample:
John
21
false
4.3

There are several ways to name variables. In Java, it's a common practice to start variable names with a lower case. If a variable name includes multiple words the first word should be in lowercase and each following word should start with an upper case.

```

1 public class Main {
2
3     public static void main(String[] args) {
4         //correct variable declarations
5         int name = "John";
6         String age = 21;
7         double isGraduated = false;
8         boolean averageScore = 4.3;
9         //do not change this code
10        System.out.println(name);
11        System.out.println(age);
12        System.out.println(isGraduated);
13        System.out.println(averageScore);
14    }
15 }
16

```

FIGURE 9
Sololearn guided inquiry.

not able to answer correctly then “0” will be awarded to that student. In Sololearn, no specific hints are provided by the application. However, guidance through text is provided in Sololearn as depicted in Figures 8, 9.

For the calculation of marks of the students, researchers calculated the attempts of the students for solving inquiry-type questions. If a student finishes in the first attempt, then “3” marks will be awarded to that student. If a student makes two attempts to solve then “2” marks will be given. Similarly, if a student answers the question on the third attempt, then a “1” mark will be awarded. However, if a student makes more than 3 attempts to solve the given question then a “0” mark will be given to that student. The following sections represent the results of both groups at this evaluation level.

For structured and guided inquiries, students of both applications; MILOS and Sololearn were asked to answer 8 questions comprising 24 scores related to the topic “Variables.” An independent sample *t*-test was conducted on the marks obtained by both groups; Experimental and Control, to identify significant differences in their learning performances. However, the open inquiry result of MILOS was not considered for this independent sample *t*-test. According to the *t*-test result as shown in Table 5, there is a statistically significant difference ($p < 0.01$) between marks obtained by the experimental (MILOS) group students as compared to the control (Sololearn) group students. The results indicated that students performed much better using MILOS as they achieved an average of 82.7% (19.85 out of 24) marks. In contrast, the control (Sololearn) group students only achieved

TABLE 5 Independent sample *t*-test of experimental and control groups.

Group	Application participants (N)	Mean (M)	Standard deviation (SD)	Standard error (SE)
Experimental (MILOS)	54	19.85	1.84	0.25
Control (Sololearn)	55	15.76	2.81	0.38

65.68% average marks. Furthermore, the research findings have practical significance as the effect size ($d = 1.72$) between experimental and control groups are large ($d > 0.8$).

Comparing the learning performances of these two groups, inquiry-type wise marks were evaluated. At each inquiry level, 4 questions were asked from both groups except in the open inquiry type, which was evaluated by the experimental group only. In an open inquiry, only 2 questions were asked from students. According to the results, both groups performed better in the structured inquiry type as compared to the guided inquiry type. Overall, experimental group students performed much better in both types of inquiries as illustrated in Figure 10. It has been further indicated that the open inquiry type is more challenging to the experimental group students in comparison with other inquiry types including structured and guided.

Gender-wise results of both groups are depicted in Figures 11, 12. For the experimental group, 30 male and 24 female students

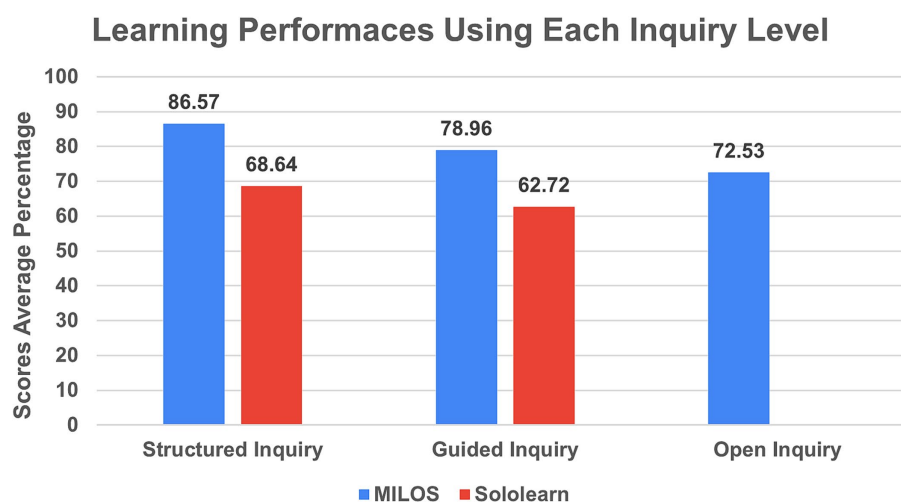


FIGURE 10

Learning performances using each inquiry type comparison between experimental and control groups.

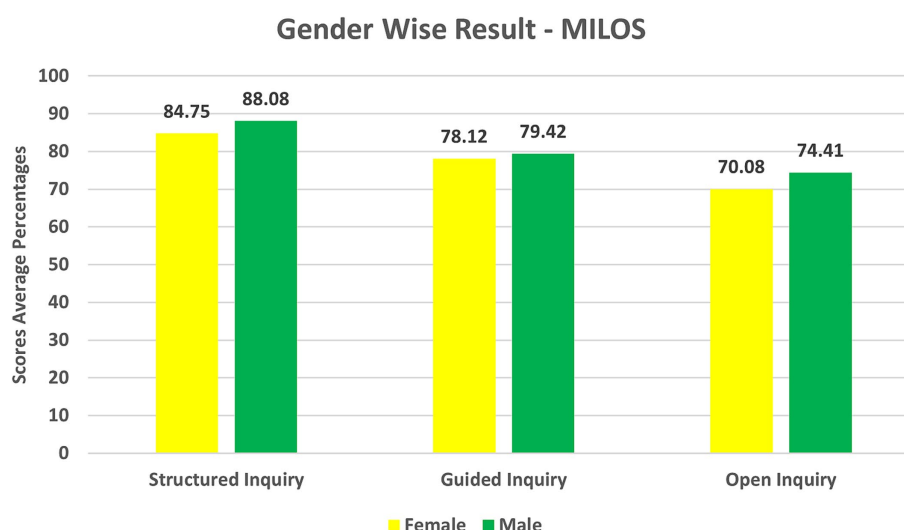


FIGURE 11

Gender-wise results—MILOS.

participated while for the control group, 32 males and 23 females answered the given questions. According to the results, there is a significant difference found in both groups that male students secured more marks in answering inquiry-type questions in comparison to female students. The results in percentages confirm these explanations as shown in Figures 11, 12. The possible reasons for discrepancy among male and female students may be because of the understanding the programming concepts and the use of such applications with ease as few of these participants discussed during micro level evaluations.

The correlation between both variables technological usability and learning performance among participants of both groups are also calculated. According to the results, the technological usability and learning performance of the experimental group are positively correlated ($r = 0.8617$). Similarly, the correlation between technological usability and learning performance among the control

group participants are ($r = 0.8136$). This indicates both variables have a strong positive correlation among both groups.

6 Discussion

The results and analyses of this research demonstrate that the use of multiple inquiry levels in a programming course can foster motivation and engagement in learning about programming concepts. They further highlight that this research may improve students' inquiry and programming skills through multiple inquiry levels. In addition, the experimental (MILOS) group outperformed the control (Sololearn) group in such inquiry activities.

There are few studies (Martín-Ramos et al., 2018; Tsai, 2019; Gunis et al., 2020; Venant et al., 2017) found in the literature where a

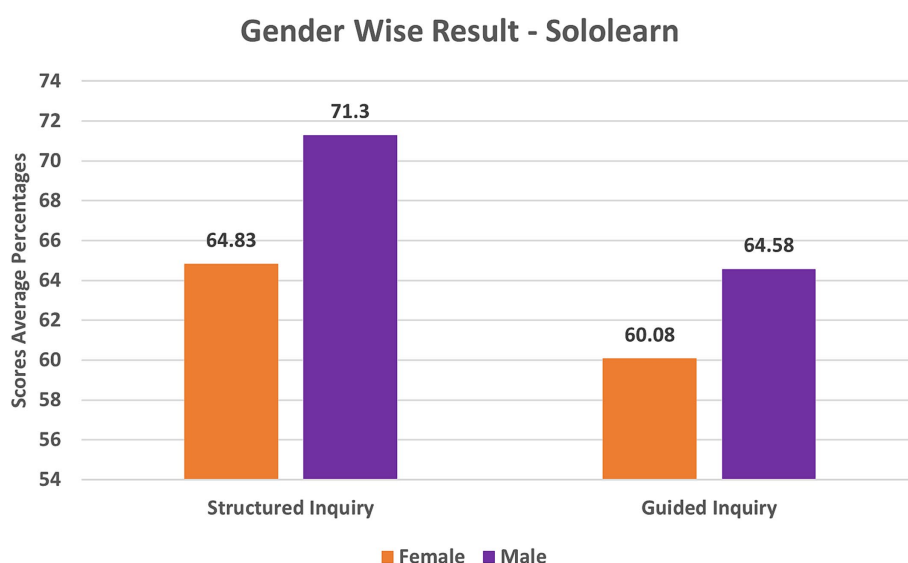


FIGURE 12
Gender-wise results—Sololearn.

single level of inquiry level was used for learning programming concepts. For instance, Tsai (2019) highlighted that inquiry-based learning in a programming environment creates cognitive challenges that can enhance students' motivation toward learning programming skills. Further, visual programming language guides them to exhibit students' problem-solving and inquiry skills.

jeKnowledge (Martín-Ramos et al., 2018), an application was designed using a peer-to-peer strategy for high-school and first-year undergraduate students to learn introductory programming courses based on the Arduino open-source platform. This application guides students to learn the domain knowledge and enhances their motivation toward computational thinking skills. The pre- and post-surveys indicated that first-year university students possess more interest compared to high school students. In addition, there is a significant difference found in learning programming concepts among male students in comparison with female students. Our results also supported these findings where male students have outperformed as compared to female students while learning programming courses in software engineering education. In another instance, an inquiry-based learning approach was used to teach Python programming at the secondary school level (Gunis et al., 2020). In this research, teachers observed that students were not accustomed to inquiry-based learning. However, with time, they understand this approach and enjoy learning programming through it. Further, they indicated that this structured inquiry approach provides a significant impact on developing students' computational thinking skills. Nevertheless, on some occasions, students tend to take more time if they do not have basic programming skills.

In a similar context, a web-based virtual learning environment named Lab4CE (Laboratory for Computer Education) was evaluated by students to access their programming skills (Venant et al., 2017). In this study, students learn Shell programming commands through the Lab4CE learning environment. One of the quantitative indicators of this study was the guided inquiry while learning Shell programming commands. However, the results indicated a weak significant

correlation with assessment scores. Researchers mentioned that this may be due to the reasons as the learning environment did not cater to the way students sought guidance (i.e., after a command failure, testing a new command, etc.) (Venant et al., 2017).

In contrast, Sololearn was the only study in which multiple levels of inquiries; structured and guided were used to reinforce object-oriented programming (OOP) concepts taught in their classrooms (Deshmukh and Raisinghani, 2018). This application was evaluated by students to access their programming skills through collaborative inquiry-based learning while understanding OOP concepts (Deshmukh and Raisinghani, 2018). According to these results, 76% were satisfied with the Sololearn application, however, 14% indicated that there is a need to guide through a better explanation of the underlying concepts. The remaining 10% were not satisfied at all. However, no learning performance was measured using Sololearn in this research. On the other hand, our findings have also indicated the same that 71% of students (3.54 out of 5 in Q6 mentioned in Table 4) have a positive attitude toward using Sololearn. In addition, the average percentage of marks obtained by students using Sololearn is 66%. These results further validated the given interpretations.

7 Conclusions and future work

This research explores the use of multiple inquiry levels in a programming taught course of software engineering education. For this purpose, an application, MILOS was designed using AODM. This application not only engages students to learn about an object-oriented topic "Variables" but also improves their multiple inquiry level skills including structured, guided, and open. Further, it demonstrates that students using MILOS have performed better in comparison with those students who used Sololearn. This research has twofold contributions, which were sparsely investigated in the previous literature; the practical implementation of multiple inquiry levels in a programming course in software engineering education

and exploring the performance of such a learning environment through another state-of-the-art environment in this domain (Sololearn).

In this research, two research questions were answered. The first research question was answered through the design of MILOS for performing multiple inquiry levels in a programming course. The detailed design of MILOS is discussed in Section 4. For answering research question 2, both applications MILOS and Sololearn are evaluated through *Micro* and *Meso* levels of the M-3 evaluation framework. During the *Micro*-level evaluation, students' responses indicated that the experimental (MILOS) group has considerably better usability and quality aspects than the control (Sololearn) group while performing multiple inquiry levels in a programming course. As far as the *Meso* level evaluation is concerned, the experimental group significantly achieved better results as compared to the control group. The inquiry-type results also confirmed these interpretations. It also indicates that students faced more challenges while performing an open inquiry than in structured and guided inquiries. In addition, gender-wise results from both groups were also collected. According to gender-wise results, male students have performed better than female students in both groups. The above-discussed results are encouraging; however, there are some limitations to this research. The evaluation of these applications MILOS and Sololearn were conducted with the students of the department of software engineering of a local University only (Bahria University Karachi Campus), which may be repeated in some other Universities to validate these results. Only Sololearn application is compared with a designed system (MILOS) in this research because no other application is available that may consider multiple inquiry levels for learning programming skills. The evaluations with other applications may provide different results.

In the future, a new version of MILOS may be developed and evaluated for investigating multiple inquiry levels in a collaborative environment where a group of students can discuss and perform inquiries. This application only considers two levels of the M-3 evaluation framework: *Micro* and *Meso*. *Macro*-level may be evaluated to assess the long-term impact of such applications on students' inquiry learning. For such long-term evaluation, this application may further be modified while considering students' profiling to analyze students' complete course records. This designed application can be part of the curricula by using it in the computer laboratory for guiding programming students through multiple inquiry levels.

Data availability statement

The original contributions presented in the study are included in the article/[Supplementary material](#), further inquiries can be directed to the corresponding author.

Ethics statement

The studies involving humans were approved by Departmental Ethics Committee, Bahria University Karachi, Pakistan. The studies were conducted in accordance with the local legislation and

institutional requirements. The participants provided their written informed consent to participate in this study.

Author contributions

SA: Conceptualization, Methodology, Supervision, Writing – original draft, Writing – review & editing. SB: Conceptualization, Data curation, Investigation, Software, Writing – original draft. AA: Formal analysis, Methodology, Writing – review & editing. OR: Project administration, Visualization, Writing – review & editing. FA: Formal analysis, Visualization, Writing – review & editing. KA: Methodology, Validation, Writing – review & editing. TL: Project administration, Validation, Writing – review & editing.

Funding

The author(s) declare that no financial support was received for the research and/or publication of this article.

Acknowledgments

We acknowledge the support of the faculty members of the Department of Software Engineering, Bahria University, Karachi Campus who spare students to conduct this research. We also acknowledge the support provided by the Head of the Department, Software Engineering, in conducting this research.

Conflict of interest

SB was employed by Habib Bank Limited, Pakistan.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The authors declare that no Gen AI was used in the creation of this manuscript.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/feduc.2025.1503996/full#supplementary-material>

References

- Ahmed, S., Javadi, S., Niazi, M. F., Alam, A., Ahmad, A., Baig, M. A., et al. (2019). A qualitative analysis of context-aware ubiquitous learning environments using Bluetooth beacons. *Technol. Pedagogy Educ.* 28, 53–71. doi: 10.1080/1475939X.2018.1557737
- Ahmed, S., and Parsons, D. (2013). Abductive science inquiry using mobile devices in the classroom. *Comput. Educ.* 63, 62–72. doi: 10.1016/j.compedu.2012.11.017
- Ahmed, S., Parsons, D., and Mentis, M. (2012a). Scaffolding in Mobile science enquiry-based learning using ontologies. *ICST Trans. e-Educ. e-Learn.* 12:e5. doi: 10.4108/eeel.2012.07-09.e5
- Ahmed, S., Parsons, D., and Mentis, M. (2012b). “An ontology supported abductive Mobile enquiry based learning application” in 2012 IEEE 12th international conference on advanced learning technologies (Rome, Italy: IEEE), 66–68.
- Banchi, H., and Bell, R. (2008). The many levels of inquiry. *Sci. Child.* 46, 26–29. doi: 10.3217/jucs-019-14-2093
- Bybee, R. W. (2019). GUEST editorial: using the BSCS 5E instructional model to introduce STEM disciplines. *Sci. Child.* 56, 8–12. doi: 10.2505/4/sc19_056_06_8
- Chao, P. Y. (2016). Exploring students’ computational practice, design and performance of problem-solving through a visual programming environment. *Comput. Educ.* 95, 202–215. doi: 10.1016/j.compedu.2016.01.010
- Chaudhary, Z., and Lam, G. Embracing the unexpected: using troubleshooting to modulate structured-to-open-inquiry and self-regulated learning in a first-year virtual lab. Proceedings of the Canadian Engineering Education Association (CEEAA). (2021).
- Chua, B. B., and Dyson, L. E. (2004). “Applying the ISO 9126 model to the evaluation of an e-learning system” in Proceedings of the 21st ASCILITE conference. eds. R. Atkinson, C. Mc Beath, J. Jonas-Dwyer and R. Phillips, 184–190.
- Deshmukh, S. R., and Raisinghani, V. T. (2018). “ABC: application based collaborative approach to improve student engagement for a programming course” in 2018 IEEE tenth international conference on Technology for Education (T4E) (Chennai, India: IEEE), 20–23.
- Gunis, J., Snajder, L., Tkacova, Z., and Gunisova, V. (2020). “Inquiry-based python programming at secondary schools” in 2020 43rd international convention on information, communication and electronic technology (MIPRO) (Opatija, Croatia: IEEE), 750–754.
- Jong, M. S., and Tsai, C. C. (2016). Understanding the concerns of teachers about leveraging mobile technology to facilitate outdoor social inquiry learning: the EduVenture experience. *Int. Learn. Environ.* 24, 328–344. doi: 10.1080/10494820.2015.1113710
- Kallas, K., and Pedaste, M. (2018). “Mobile-based inquiry learning application for experiment planning in the 8th grade chemistry classroom” in Proceedings of the 26th international conference on computers in education, 387–396.
- Khan, B. F., and Ahmed, S. (2025). An agile approach for collaborative inquiry-based learning in ubiquitous environment. *Int. J. Adv. Comput. Sci. Appl.* 16, 323–333. doi: 10.14569/IJACSA.2025.0160132
- Ladachart, L., Chaimongkol, J., and Phothong, W. (2022). Design-based learning to facilitate secondary students’ understanding of pulleys. *Aust. J. Eng. Educ.* 27, 26–37. doi: 10.1080/22054952.2022.2065722
- Lin, F. T., and Lin, Y. C. (2017). “An ontology-based expert system for representing cultural meanings: an example of an art Museum” in 2017 Pacific neighborhood consortium annual conference and joint meetings (PNC) (Tainan, Taiwan: IEEE), 116–121.
- Liu, C., Zowghi, D., Kearney, M., and Bano, M. (2021). Inquiry-based mobile learning in secondary school science education: a systematic review. *J. Comput. Assist. Learn.* 37, 1–23. doi: 10.1111/jcal.12505
- Martin-Ramos, P., Lopes, M. J., Lima da Silva, M. M., Gomes, P. E. B., Pereira da Silva, P. S., Domingues, J. P. P., et al. (2018). Reprint of ‘first exposure to Arduino through peer-coaching: impact on students’ attitudes towards programming’. *Comput. Hum. Behav.* 80, 420–427. doi: 10.1016/j.chb.2017.12.011
- Medeiros, R. P., Ramalho, G. L., and Falcao, T. P. (2019). A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Trans. Educ.* 62, 77–90. doi: 10.1109/TE.2018.2864133
- Mikroyannidis, A., Okada, A., Scott, P., Rusman, E., Specht, M., Stefanov, K., et al. (2013). WeSPOT: a personal and social approach to inquiry-based learning. *J. Univ. Comput. Sci.* 19, 2093–2111. doi: 10.3217/jucs-019-14-2093
- Muskita, M., Subali, B., and Djukri, D. (2020). Effects of worksheets base the levels of inquiry in improving critical and creative thinking. *Int. J. Instruct.* 13, 519–532. doi: 10.29333/iji.2020.13236a
- Mwanza-Simwami, D. (2011). AODM as a framework and model for characterizing learner experiences with technology. *J. E-Learn. Knowl. Soc.* 7, 75–85. doi: 10.20368/1971-8829/553
- Mwanza-Simwami, D. (2017). “Fostering collaborative learning with mobile web 2.0 in semi-formal settings” in Blended learning: Concepts, methodologies, tools, and applications. eds. M. Khosrow-Pour, S. Clark, M. E. Jennex, A. Becker and A. Anttiroiko, 97–114.
- Papasalouros, A., and Chatzigiannakou, M. (2018). Semantic web and question generation: An overview of the state of the art. In: Proceedings of the International Conference on e-Learning. Madrid, Spain, 189–192.
- Pertiwi, A. P. (2018). MOOC advancement: From desktop to Mobile phone. An examination of Mobile learning practices in mobile massive open online course (MOOC). Gothenburg, Sweden: University of Gothenburg.
- Rahmat, I., and Chanunan, S. (2018). Open inquiry in facilitating metacognitive skills on high school biology learning: an inquiry on low and high academic ability. *Int. J. Instruct.* 11, 593–606. doi: 10.12973/iji.2018.11437a
- Tan, E., Rusman, E., Firssova, O., Ternier, S., Specht, M., Klemke, R., et al. (2018). “Mobile inquiry-based learning: relationship among levels of inquiry, learners’ autonomy and environmental interaction” in Proceedings of 17th world conference on Mobile and contextual learning. eds. D. Parsons, R. Power, A. Palalas, H. Hambrock and K. Mac Callum, 22–29.
- Tsai, C. Y. (2019). Improving students’ understanding of basic programming concepts through visual programming language: the role of self-efficacy. *Comput. Hum. Behav.* 95, 224–232. doi: 10.1016/j.chb.2018.11.038
- Tsai, C. Y., Shih, W. L., Hsieh, F. P., Chen, Y. A., and Lin, C. L. (2022). Applying the design-based learning model to foster undergraduates’ web design skills: the role of knowledge integration. *Int. J. Educ. Technol. High. Educ.* 19:4. doi: 10.1186/s41239-021-00308-4
- Umer, M., Nasir, B., Khan, J. A., Ali, S., and Ahmed, S. (2017). “MAPILS: Mobile augmented reality plant inquiry learning system” in 2017 IEEE global engineering education conference (EDUCON). (Athens, Greece: IEEE), 1443–1449.
- Vavoula, G., Sharples, M., Rudman, P., Meek, J., and Lonsdale, P. (2009). Myartspace: design and evaluation of support for learning with multimedia phones between classrooms and museums. *Comput. Educ.* 53, 286–299. doi: 10.1016/j.compedu.2009.02.007
- Venant, R., Sharma, K., Vidal, P., Dillenbourg, P., and Broisin, J. (2017). “Using sequential pattern mining to explore learners’ behaviors and evaluate their correlation with performance in inquiry-based learning” in 12th European conference on technology enhanced learning (EC-TEL 2017), vol. 10474 (Tallinn).
- Vinnervik, P. (2023). An in-depth analysis of programming in the Swedish school curriculum—rationale, knowledge content and teacher guidance. *J. Comput. Educ.* 10, 237–271. doi: 10.1007/s40692-022-00230-2
- Wang, X. M., and Hwang, G. J. (2017). A problem posing-based practicing strategy for facilitating students’ computer programming skills in the team-based learning mode. *Educ. Technol. Res. Dev.* 65, 1655–1671. doi: 10.1007/s11423-017-9551-0
- Yi-Ming Kao, G., and Ruan, C. A. (2022). Designing and evaluating a high interactive augmented reality system for programming learning. *Comput. Hum. Behav.* 132:107245. doi: 10.1016/j.chb.2022.107245