Factors Influencing Programming Expertise in a Web-based E-learning Paradigm

Wajid Rafique and Wanchun Dou
Department of Computer Science and Technology, Nanjing University
State Key Laboratory of Novel Software Technology, Nanjing University

Khalid Hussain
School of Business, East China University of Science and Technology, Shanghai

Khurshid Ahmed
School of Information Management, Nanjing University, Nanjing

Abstract

Modern internet technologies have revolutionized traditional education by providing flexible and resourceful e-learning opportunities in all fields of life. Programming is an integral part of the undergraduate curriculum in computer sciences where an adequate level of programming expertise is expected from the graduates. In this paper, we explore and examine the key factors that contribute to developing programming skills among undergraduate students in e-learning. We propose that programming education follows the Technology Acceptance Model (TAM), which affects the students' attitude toward learning. We extend the TAM by integrating the factors of teaching practices, intrinsic factors, perceived usefulness, and efficacy problems with the learning intentions in our research framework.

This research involves the responses of the 460 final year students studying for a Bachelor of Computer Science and Software Engineering at an e-learning institution. Structural Equation Modelling (SEM) and Confirmatory Factor Analysis (CFA) have been employed to evaluate the relationship between factors of the model. Experimental results demonstrate that teaching practices, intrinsic factors, and perceived usefulness play a key role in endorsing learning intentions in the students. Further analysis reveals that learning intentions positively influence the programming expertise whereas an adverse impact has been observed from the efficacy problems. The results proclaim that perceived usefulness, teaching practices, and intrinsic factors develop adequate learning intentions in the students which overcome the efficacy problems and lead to better programming expertise. This research provides critical implications for policymakers to effectively implement computer science programs in an e-learning paradigm.

Keywords: e-learning, programming, web, expertise, teaching, programming education, barriers in programming.

Rafique, W., Dou, W., Hussain, K., & Ahmed, K. (2020). Factors influencing programming expertise in a web-based e-learning paradigm. *Online Learning, 24*(1), 162-181. https://doi.org/10.24059/olj.v24i1.1956

Factors Influencing Programming Expertise in a Web-based E-learning Paradigm

E-learning has been extensively introduced in higher educational institutions due to the rapid development in information and communication infrastructure (Hung & Chou, 2015). Many universities have started to offer online degree courses in addition to traditional study programs. Online services provide immense opportunities for effective implementation of e-learning. They offer the flexibility for e-learning providers to host their Learning Management Systems (LMS) online. Moreover, it enables students to access the course material independent of location and time constraints (Jose & Christopher, 2018). Additionally, web technologies provide the ability to support big data and multimedia streams that provide independence to the e-learning providers from bandwidth limitations, computation resources, storage concerns, and many other issues. Figure 1 explains the architecture of a web-based e-learning education system. It illustrates that the e-learning stakeholders interact with a web-based e-learning management system equipped with virtual machines and physical hardware. Students, tutors, and the administrators interact with the LMS using the interface provided by the service providers. LMS stands at the core of e-learning as it provides all the educational interactions to the students.

Due to a wide proliferation of web technologies, a huge number of free online courses are available hosted by YouTube, Coursera, Udemy, Edx, and many others. A wide range of e-learning courses has been available in every field of life including history, social sciences, natural sciences, engineering, and medicine. Computer science has been one of the most popular education disciplines because of an ever-increasing demand for IT professionals. Programming is the fundamental aspect of computer science programs where most of the universities start computer science curriculum with the programming courses all over the world (Raigoza, 2017).

An adequate level of programming expertise has been expected from the graduates in computer sciences. Programming education is demanding because it involves logical reasoning, mathematical skills, and extensive domain knowledge; moreover, it becomes more challenging in e-learning (Lam, Chan, Lee, & Yu, 2008). Programming education requires extensive efforts from students because they must solve complex program logic and develop a procedural algorithm to develop the code for the underlying problem. Because of these complexities, a higher dropout ratio has been observed in the computer science degree programs (Sarpong, Arthur, & Amoako, 2013). Similarly, a huge number of students complete their undergraduate studies in computer science by getting a sufficient amount of general knowledge in programming but lacking in specific skills to develop high-quality computer applications (Kelleher & Pausch, 2005).

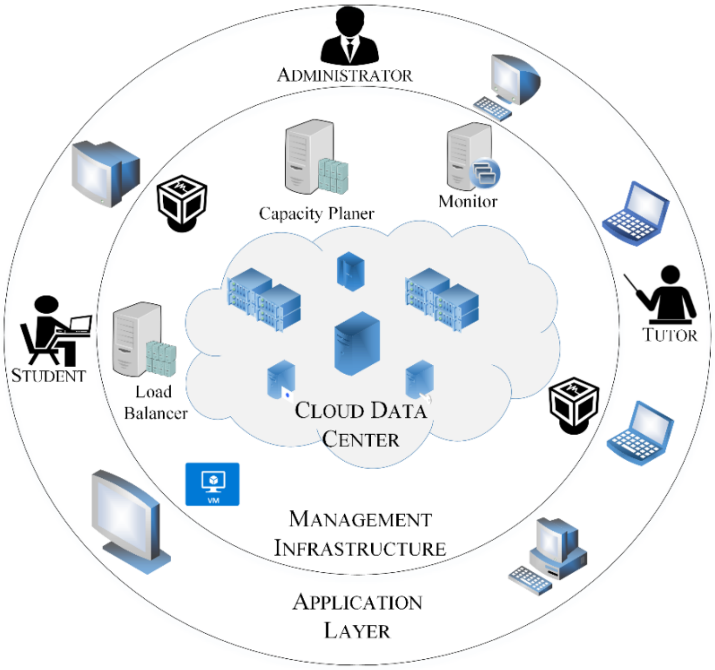

Figure 1. A web-based e-learning system model.

In e-learning, the students become frustrated when they try to execute their code without a correct understanding of the problem and forcing the entire program to run without modular development and testing (Raigoza, 2017). Therefore, programming education in e-learning becomes a critical challenge. Previous studies in this context, mainly focus on the motivational factors that affect programming education in computer science students (Law, Lee, & Yu, 2010). Although a wide range of research is available in determining problems faced by e-learning students in general, there is a lack of literature available on identifying issues in programming education. Hence, it becomes a noticeable challenge for the practical envisioning of the e-learning (Liaw, 2008). To address these challenges, we aim at designing this study to identify and analyze the fundamental factors that involve in the development of programming skills in e-learning. We perform the analysis from the perspective of both the student and the tutoring environment. A holistic approach has been used in programming education ecosystem to identify the core factors that affect programming skills development in e-learning. Moreover, the interdependence of these factors has been analyzed. This research can be implemented to facilitate policy makers and administrators to effectively develop, deliver, and manage e-learning in solving the problem of programming education. The main contributions of this study are as follows:

- We perform an extensive literature study and identify Teaching Practices (TP), Intrinsic Factors (IF), Perceived Usefulness (PU), Efficacy Problems (EP), and Learning Intentions (LI) as key factors contributing toward programming expertise development in e-learning.
- A research model has been proposed by extending the Technology Acceptance Model (TAM), which measures the dependency of these factors on overall programming skills development.
- A set of detailed validation and evaluation experiments have been performed on the survey-based data to demonstrate the data validation. Moreover, Structural Equation Modelling (SEM) and Confirmatory Factor Analysis (CFA) have been employed to evaluate the effectiveness of the proposed research model.

Review of Literature

Computer science constitutes one of the most important degree programs offered by e-learning institutions due to an ever-increasing demand for IT professionals. Programming is the fundamental skill expected from computer science graduates (Lam et al., 2008). It requires analytical and problem-solving skills from the learner that involve describing processes and procedures, developing algorithms, and the implementation in the desired programming language (Law et al., 2010). E-learning students lose interest and face problems during the coding tasks; therefore, there is a strong need to improve programming education in the e-learning. We perform an extensive literature survey and find that five factors greatly affect programming expertise in e-learning, which includes TP, IF, EP, PU, and LI. We devise a research model by extending TAM and explain how these factors influence the programming expertise in e-learning.

Nganji (2018) reveals that education providers must focus on learners to increase their participation in the learning process, which can improve their knowledge and skills. Therefore, the medium of instruction plays an important role in the learning process. The communication strategy (e.g., synchronous, asynchronous) strongly influences the students' understanding of the subject. In synchronous communication, direct interaction among teachers and students provide the basis for the academic discussions that help the students assimilate the course content. Alternatively, asynchronous communication is adopted in the e-learning where the discussion forums and emails are used for the student-teacher interaction. Offir, Lev, and Bezalel (2008) demonstrate that asynchronous communication yields an adverse impact on the performance of the students. They propose that asynchronous communication does not produce a student-teacher dialogue that deprives the students of asking questions. Boelens, De Wever, and Voet (2017) devise a strategy to arrange face-to-face meetings at the start of the course so that students get an introduction of their mentors and their classmates. This introduction provides the e-learning students a sense of community later in the course. The impact of student-teacher interaction on the final year project has been discussed by Dos Santos and Cechinel (2018); their findings reveal that face-to-face meetings yield positive results. Hence, the student-teacher interaction is at the core of e-learning for effective implementation.

Programming is challenging in a way that it requires both a theoretical understanding of the concepts and hands-on experience in specific programming languages (Lam et al., 2008). Numerous techniques have been devised to facilitate the complex programming environment including pair programming, shared code, and tools to facilitate the debugging process. Sarpong et al. (2013) suggest that extensive lab work under the guidance of a tutor for programming tasks help students to master programming skills and decrease the retention rates. Celepkolu and Boyer (2018) discuss the importance of a shared coding system in a hybrid pair programming environment to overcome the common mistakes performed by the students. Zin, Idris, and Subramaniam (2006) introduce a virtual pair programming solution where one student performs the coding and the other proofreads simultaneously. However, this technique consumes a lot of time; moreover, it requires constant interaction between the students, which sometimes becomes difficult in e-learning. Lam et al. (2008) propose an automatic debugger to solve the problem of the mentor providing hints on the mistakes during the coding process. Students submit their code to a debugger which identifies common errors and offers suggestions for improvement; however, this tool only works for smaller programming tasks.

One of the primary medium of communication in e-learning is LMS, which facilitates the student-teacher interaction; moreover, it enables effective follow-up of the course activities.

Yunkul and Cankaya (2017) present the importance of Edmodo LMS which provides a secure environment for student-teacher interaction and feedback. It also incorporates the social media platform with LMS that creates a social environment among the students. Ateş Çobanoğlu (2018) explores the preferences of students in an information technology course to learn in a blended learning environment, which involves both traditional education and e-learning. Their results proclaim that the use of LMS for blended learning increased student's performance. Similarly, the usefulness of the underlying e-learning course for a student is also an important factor in achieving programming skills. B.-C. Lee, Yoon, and Lee (2009) suggest that the students who perceive that the e-learning course is beneficial in their future try to perform better during their studies.

Different solutions have been proposed for solving problems faced by the students during programming. However, there is still a lack of research available on ascertaining factors which involve programming education. The proposed tools in the research facilitate the specific aspects of programming education; however, a holistic approach in the programming education ecosystem is still not available. Therefore, we provide an empirical study and design a research framework for the identification and analysis of factors that affect programming expertise in e-learning.

E-learning Research Framework

We present a research framework after an extensive literature review and identify TP, IF, EP, PU, and LI as the key factors involved in programming skills development in e-learning (Martín-Rodríguez, Fernández-Molina, Montero-Alonso, & González-Gómez, 2015). E-learning acceptance follows TAM (Venkatesh & Davis, 2000) for effective implementation in different countries. After a thorough review of the literature, it has been established that TAM offers the key factors for effective acceptance of e-learning by the students. In the TAM framework, the flexibility exists for adding more factors (variables) depending on the contextual scenario (Pituch & Lee, 2006). Similarly, TAM predicts an individual's attitude toward using ICT technologies and it owns a widespread background in the field of e-learning (Alharbi & Drew, 2014). The use of TAM toward the perceptions of the teachers while teaching online has been studied by Wingo, Ivankova, & Moss, 2017. The EP factor in our research framework moderates the programming expertise, which is strongly associated with the Perceived Ease of Use (PEU) factor in TAM where the problems in PEU can negatively affect the intentions to use. The intention factor in TAM corresponds to the LI in our framework, which demonstrates the motivation to learn programming. Finally, the usage behavior in the TAM assimilates to the programming expertise in our framework which corresponds to the effectiveness in adapting to the programming education in e-learning. A critical review of the research in the field of e-learning demonstrates that TAM has not yet been studied in the e-learning paradigm to learn computer sciences. Therefore, we utilize PU and IF to predict the behavioral intention of programming education, moreover, we add TP to develop those behavioral intentions which are aligned with the TAM framework requirements. Furthermore, we add computer self-efficacy in performing programming tasks to show the impact on the programming expertise of students. Due to the above-mentioned reasons, we adopt TAM to ascertain the impact of students' intentions toward the continuous use of e-learning for programming expertise development. Figure 2 illustrates the configuration of all the factors in the framework. We discuss the development of the research hypothesis below.

3.1 Learning Intentions (LI)

LI can be defined as the extent to which continuous effort has been directed toward achieving a specific goal (here the goal corresponds to learning how to code effectively).

Programming requires constant effort where LI is comprehended by the intuition of students to learn and practice the programming tasks (Xia & Liitiäinen, 2017). LI depends on the effectiveness of TP, the PU of the studies, and IF to study in an e-learning course (Pugh, 2019). As discussed in the literature review, that TP greatly enhances the LI of the students, hence students perform well in understanding the concepts through TP. These motives serve as an impetus for learning plans that supports students in knowledge acquisition and academic success (Linnenbrink & Pintrich, 2002). The quality of the education system plays an essential role in determining the success of the students moreover, the mentoring process and support increase the LI of the students (Dorner & Kárpáti, 2010). Hence, as this study proposes that LI positively impact the programming expertise in e-learning, we formulate the hypothesis H3.

H3: LI positively impact the programming expertise of students in an e-learning system.

3.2 Intrinsic Factors (IF)

IF constitute the motivation of a student toward the learning process; moreover, it involves the individual's personal rather than environmental setting (Hendijani, Bischak, Arvai, & Dugar, 2016). Students need to be motivated in e-learning because it is easy to lose self-evaluation in the state of isolation (Galusha, 1998). Khan and Nawaz (2013) argue that when intrinsic motivations are high, learning outcomes are positive, which demonstrates that IF play a positive role in developing LI. Most of the IF in higher education involve students' satisfaction in the current studies and their interest in the current course (Bouhnik & Marcus, 2006). When students are satisfied with their studies, they learn effectively, which yields a positive impact on their overall skills development in e-learning (Eom, Wen, & Ashill, 2006). Considering the above discussion, we hypothesize that the effect of IF is positively related to the LI. From this discussion, we devise hypothesis H1 and H6.

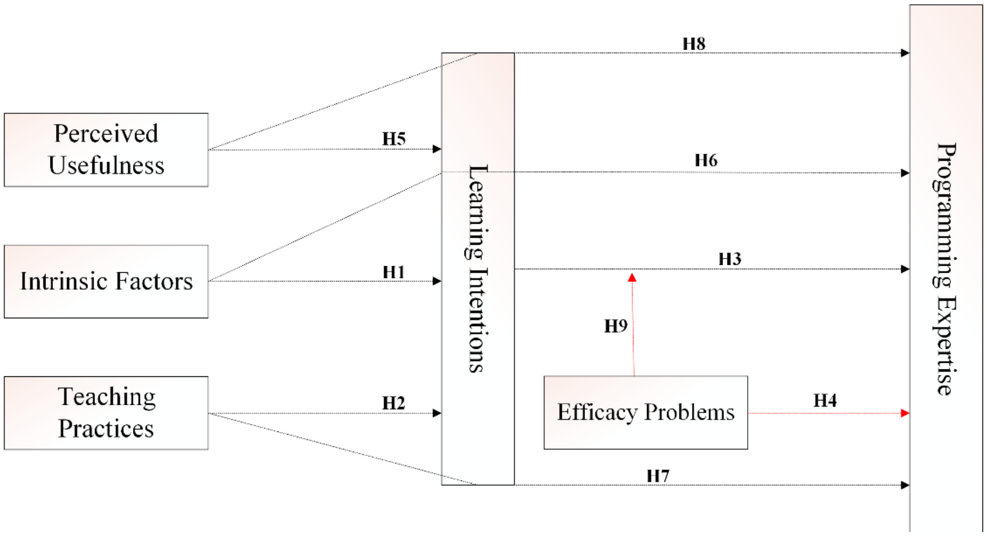

Figure 2. E-learning research model indicating all the hypothesis of the research.

- **H1:** IF positively affect the student's intentions toward programming education.
- **H6:** The impact of IF on LI positively transcend toward the programming expertise via an indirect passage of LI.

3.3 Teaching Practices (TP)

TP constitutes the teaching methodologies adopted in the e-learning (Martin, Budhrani, Kumar, & Ritzhaupt, 2019). In e-learning, TP involves the course content delivery methods including video lectures, handouts, and LMS discussions for mentoring (Valentine, 2002). Students perceive that programming is effective and enjoyable in the presence of a mentor providing hints on their mistakes. However, it is difficult to provide such mentors in e-learning. Offir et al. (2008) posits that direct interaction between students and the teachers positively affect the students' learning process, which helps them to discuss their problems and get immediate feedback from their mentors. TP greatly affects the programming expertise of students because a positive correlation exists between the attitude of a learner and the mentoring process (Dahalan, Hassan, & Atan, 2012). When students keep constant interaction with their tutor, they grasp more concepts and discuss their issues with their mentor, This results in a positive impact on students' learning process (Berry, 2019). This study categorizes TP as one of the key factors that influence programming expertise and measures the effect of TP on students' LI. Hence, we hypothesize the following.

- **H2:** TP in e-learning positively influences students' LI.
- **H7:** The impact of TP positively trends toward the programming expertise via an indirect path of LI.

3.4 Efficacy Problems (EP)

Efficacy is a self-belief to execute a course of action to attain the desired learning outcome in the e-learning system, factors that negatively impact efficacy have been denoted as EP. It is natural that the students face problems during the programming tasks hence. Immediate support can help them get out of the programming complications where they tend to plunge. Most of the times students lose interest in programming while they practice by themselves and experience failures in the learning process. Jenkins (2001) reveals that special mentoring arrangements are required to teach programming in e-learning to enhance efficacy. Allen, Cartwright, and Stoler (2002) suggest that it is difficult for some beginners to start programming. Automated Integrated Development Environments (IDEs) can be used to assist them in writing the code. The PASS program submission system has been developed to facilitate beginners learning programming, which incorporates an easy-to-use IDE to assist students in programming education (Law et al., 2010; Yu, Poon, & Choy, 2006). The isolation also contributes to EP; students feel the sense of isolation due to the non-availability of face-to-face interaction with their peers. This obscures their learning process. Taking these arguments into consideration, we hypothesize that EP negatively affects the programming expertise and moderates the influence of LI on student programming expertise. Hypothesis H4 and H9 have been formulated from this discussion.

- **H4:** EP adversely impacts programming expertise.
- **H9:** EP moderate the impact of LI on programming expertise.

3.5 Perceived Usefulness (PU)

PU means the extent to which e-learning students find their course beneficial (B.-C. Lee et al., 2009). PU has widely been used to predict the adoption of e-learning (Tarhini, Hone, & Liu, 2014; Y.-H. Lee, Hsiao, & Purnomo, 2014). It is also pertinent to note that the PU has a positive influence in developing the behavioral intention of the students (Jan & Contreras, 2011). When the students find that their studies are beneficial to them in the future they put in more effort; hence, their LI increase (Cheng, 2011). Keeping this in mind, we hypothesize the following.

- **H5:** PU has a positive impact on developing LI in programming education.
- **H8:** The impact of PU on LI trends positively toward programming expertise in e-learning.

Methods

Figure 3 describes the workflow of current research, which includes extensive background study, factors identification, questionnaire development, data collection, validation acceptation, and finally the results and discussion. The participants in this study include students of the Virtual University of Pakistan (VU) that employs a web-based LMS to facilitate e-learning. The LMS enables students to submit assignments, check results, follow the class schedule, participate in the discussions, and contact the course tutor for the discussions. The video lectures have been recorded and delivered to the students who attend them according to the schedule provided on LMS. Every tutor regularly creates a discussion topic for each lecture where students discuss their issues and problems via text messages. We employ a random sampling procedure to select participants in the current study. The original sampling frame of this study consisted of 550 students majoring in different fields of computer science. We used Google forms to conduct the survey, which made it easy to directly import the data in the analysis tools. We distributed the online survey using emails and repeatedly sent bimonthly reminders to the participants to complete up the survey. Finally, out of the total population, we were able to collect responses from 460 respondents (response rate = 83%). We used five-point Likert-scale comprising of strongly agree, agree, neutral, disagree, and strongly disagree for all items (except item 3, 7, 8, and 10). Alternatively, we measure the items 3, 7, 8, and 10 on a five-point Likert scale of strongly satisfied, satisfied, neutral, somewhat satisfied, and not satisfied.

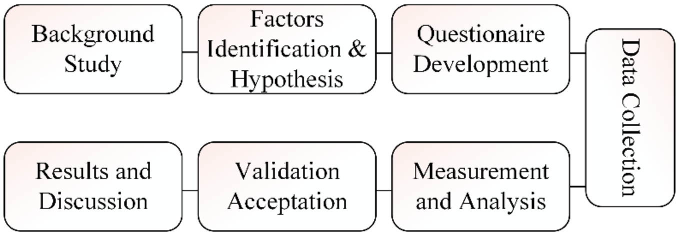

Figure 3. The workflow of research.

The data analysis was performed in the two phases. In the first phase, the demographics and the reliability study of the measurement model were performed. In the second phase, the hypothesis testing, CFA, and SEM were conducted. The research group of this study consisted of students who were in their final year of Bachelor of Computer Science (n = 309) and Software Engineering (n = 151) degree. Students in the final year develop independent projects that need extensive programming skills. The reason behind selecting these students was that they have extensive experience of studying in the e-learning system and their programming skills should have been actively developed to accomplish their final year project. The responses include students' demographics information, TP, IF, EP, LI, and programming expertise. Table 1 shows the demographic information of the students involved in the study.

Table 1.
Demographic information of the participants.

Variables		**Value**
Age	Range (years)	20-43
Gender	Male	47.8%
	Female	52.2%
Major	Computer Science	67.2%
	Software Engineering	32.8%

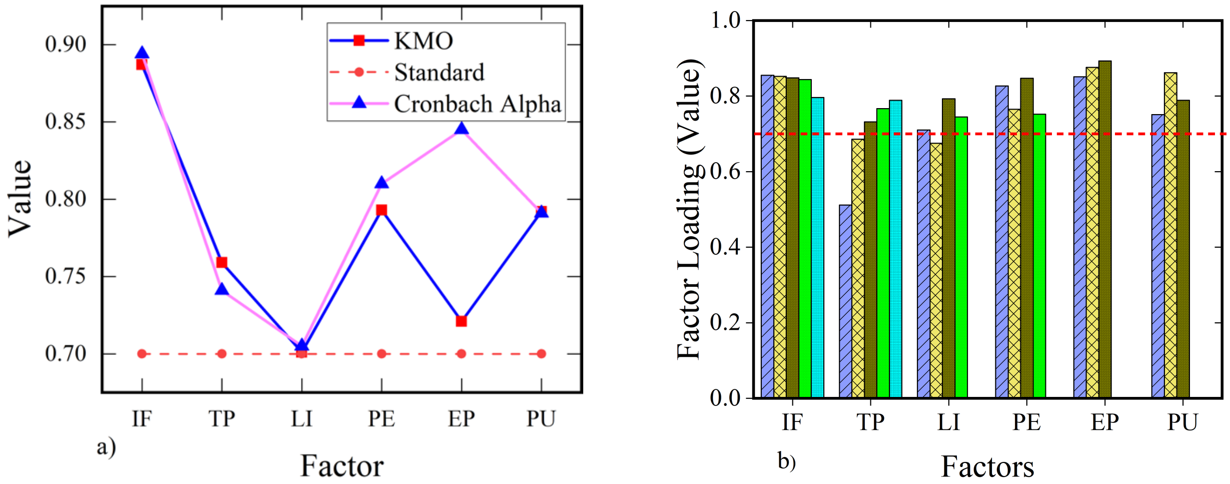

Figure 4. (a) Values of KMO and Cronbach's Alpha; (b) Factor loading for all the variables where the dotted line demonstrates the acceptable range in both the figures.

4.1 Model Validation

In this section, we present the experimental evaluation of the proposed research. We analyze the data with Statistical Package for Social Sciences (SPSS-version 22.0) and Mplus version 8.1 to measure the Exploratory Factor Analysis (EFA) and reliability coefficients (Tenenhaus, Vinzi, Chatelin, & Lauro, 2005; Chin, 1998). We initially verify data for the reliability and validity before testing the proposed hypothesis. All the measurements in this research have been newly developed. Therefore, the efficiency of these measures has been established by performing the EFA. The results of EFA are presented in Table 2, which indicates that all the measurement scales fulfill the requirements of the recommended standard. According to Hair, Black, BABIN, Anderson, and Tatham (2010), the data must be analyzed for Kaiser-Meyer-Olkin (KMO) and Bartlett's test of sphericity before proceeding to EFA. The values of KMO should be higher than 0.7 and Bartlett's trial should be significant to meet the cut-off criteria. Figure 4 elaborates the validation of the dataset; we observe from Figure 4a that the values of Cronbach's Alpha and KMO are greater than 0.7. Similarly, Figure 4b demonstrates that the values of factor loading are greater than 0.5 which verifies the suitability of the data for the current research. The result of Bartlett's test is also substantial, which shows the adequacy of the data for EFA. We evaluate the results using Principal Component Analysis (PCA), which is a highly reliable technique to measure EFA. Results in Table 2 demonstrate that the output of all the measurement scales was a single factor; moreover, the eigenvalue of the first indicator is also higher than 1.0. Hence, we choose one element for each measurement scale. Moreover, Hair et al., (2010) and Sekaran and Bougie (2011) state that the factor loadings of individual elements of each measurement scale should be higher than 0.5 to meet the criteria of convergent validity. Correspondingly, all the other items successfully loaded on the selected portions.

The factors of PU, LI, and EP are adapted from the study of the Davis and Venkatesh (1996). The IF conforms to the study of Eom et al., (2006). Similarly, TP was adapted from Hung and Chou (2015), whereas the factors of PE were extracted from Kelleher & Pausch, 2005. We used five measurement scales and factor loading of all the items on these scales was higher than 0.5, which shows the adequate convergence of the measurements. The reliability of data was measured with Cronbach's Alpha, which is the most common and statistically superior test for measuring the internal consistency of the data. Finally, Table 2 illustrates that the Cronbach's Alpha for all the measurement scales is higher than 0.7 which shows the robust reliability of measurement scales for further analysis (Hair et al., 2010; Sekaran & Bougie, 2011).

Table 2.

Statistics of EFA on the Dataset

Variables	**Item No.**	**Items**	**Factor Loading**	**KMO Value**	**Eigen Value**
Intrinsic Factors	1	Did you enroll in an e-learning institution because of its flexible accessibility?	0.855	0.887	3.521
	2	Will you like to take another higher degree course in computer sciences at an e-learning institution?	0.852		
	3	How do you rate your satisfaction level in the e-learning studies?	0.848		
	4	Did you follow-up the course material along with your other schedules regularly?	0.844		
	5	Did you find the computer science course fruitful in your career before enrolling?	0.796		
Teaching Practices	6	Do you wish to have step by step guidelines for the complex coding tasks?	0.512	0.759	2.480
	7	How do you rate the content quality of the course material?	0.686		
	8	How do you rate the lecture delivery of the tutors during the lectures?	0.732		
	9	Do you like to have face to face conversation with the tutor during coding tasks?	0.767		
	10	How effective was the instructor's response to LMS when you interact with them?	0.789		
Learning Intentions	11	Are you intending to join or already working in a software development company?	0.710	0.701	2.144
	12	Do you try to start coding and fail?	0.675		
	13	Are you motivated toward learning programming?	0.793		
	14	Do you explore online tutorials other than course material for programming help?	0.745		

Programming Expertise	15	Do you have sound knowledge of basic programming skills?	0.827	0.793	2.553
	16	Have you developed your final year project by yourself?	0.765		
	17	Do you have excellent skills in a reputed programming language, e.g., Net, Java, python?	0.847		
	18	Did you complete your programming assignments by yourself during your studies?	0.752		
Efficacy Problems	19	Do you think that if you stuck on a programming task, you are not able to get out of it, because of lack of support?	0.851	0.721	2.291
	20	Do you think that it is difficult to access right programming help on the internet?	0.876		
	21	Do you think integrated software development tools (IDE) are complex?	0.893		
Perceived Usefulness	22	Do you think that this e-learning course will earn you a good job?	0.751	0.792	2.312
	23	Do you think that you will find a good career after getting the degree?	0.862		
	24	Do you think that career-oriented learning is the need of current rapid development environment?	0.789		

4.2 Model Fitness

The hypothesized paths have been tested with the SEM technique using Mplus version 8.1. Before the SEM, we validated the data with confirmatory factor analysis (CFA). The model fitness indices for both CFA and SEM demonstrate an acceptable fit of the data with the proposed model. Five measurements of fitness indices were utilized including Comparative Fit Index (CFI), Tucker-Lewis Index (TLI), Root Mean Square Approximation (RMSEA), Standardized Root Mean Square Residual (SRMR), and chi-square to degree freedom ($\chi2$ /d.f) as presented in Table 3. Fit indices suggest that the values of CFI and TLI should be near to 0.95 for a good fit, moreover the values of RMSEA and SRMR should be less than 0.10 for an adequate fit (Hu & Bentler, 1999). Additionally, the chi-square to the degree of freedom ratio should be less than 3.0 for an acceptable fit (Amemiya & Anderson, 1990). Table 3 demonstrates the findings of the CFA and SEM indices where the values of CFI and TLI are 0.942 and 0.931, respectively. These values are aligned with the threshold described by Hu and Bentler (1999) for an acceptable fit. Additionally, the values of SRMR and RMSEA are also following the Hu & Bentler (1999) criteria of the SEM model fitness. Similarly, the value of chi-square is 1.62 which follows the criteria proposed by (Amemiya & Anderson, 1990).

Table 3.

Model Fitness Indices used in the Research.

Fitness Indices	**Values**	
	CFA	**SEM**
CFI	0.966	0.942
TLI	0.958	0.931
RMSEA	0.045	0.056
SRMR	0.048	0.058
Chi-Square $\chi2$ /d.f	1.40	1.62

To fulfill the requirements of validity and reliability, we carried out CFA of the collected data before proceeding to structural analysis. The model fit indices as presented in Table 3 indicates that the model fits well with the data. The factor loadings of individual question indicators of all the variables as shown in Table 4 are higher than 0.5, which satisfies the requirements of convergent validity (Hair et al., 2010). We deleted two question items, one from teaching practices construct and one from learning intentions scale because of lower factor loadings. The factor loading of these two items was lower than 0.5, which was not meeting the cut off criteria; hence, we carried out further analysis on the remaining items. The Cronbach's Alpha reliability test indicates that all the measures used in this study are reliable as these values are higher than 0.70 for all the constructs (Hair et al., 2010).

Table 4.

Confirmatory Factor Analysis

Construct	No. of Items	Factor Loading (Range)	Cronbach's Alpha
Intrinsic Factors	05	0.740 – 0.813	0.894
Teaching Practices	04	0.559 – 0.743	0.741
Learning Intentions	03	0.504 – 0.649	0.705
Programming Expertise	04	0.648 – 0.801	0.810
Efficacy Problems	03	0.735 – 0.880	0.845
Perceived Usefulness	03	0.613-0.793	0.791

Results

The results of the structural model have been presented in Table 5, which shows the hypothesized paths and their respective coefficients. H1 explores the relationship between IF and LI where the items measuring IF include student's intentions of joining the e-learning program, student's follow-up of the course, and intrinsic desire to learn the course. The results demonstrate that H1 has significantly been accepted, which affirms that IF has a positive impact on the LI (β = 0.743, p = 0.000) as it can also be observed in Figure 5 which shows the path performance of all the hypothesis. The plausible reason behind this is the fact that when the students join the programming course because of their interest, they follow-up the course regularly and thus are satisfied with their studies. Furthermore, their learning motivation develops higher. In the same context, H6 suggested that the impact of IF on LI positively trend toward programming expertise. This hypothesis was also supported by the results (β = 0.703, p = 0.001), it implies that students with high intrinsic motivation of joining the computer science degree programs will be able to gain good programming expertise in the future.

H2 proposes an affirmative impact of TP on LI which has been significantly revealed during the evaluation (β = 0.564, p = 0.000). The items measuring TP include help in programming tasks, course content quality, lecture delivery, and effective student-teacher interaction. Similarly, H7 recommends that the impact of TP on LI positively transcends toward the improved programming expertise (indirect effect; β = 0.937, p = 0.002). Our results provide the evidence that TP are one of the most critical factors in determining students' LI, which further contributes to programming expertise. It implies that effective TP help in developing intentions of learning, which in turn helps them in effectively grasping the programming knowledge.

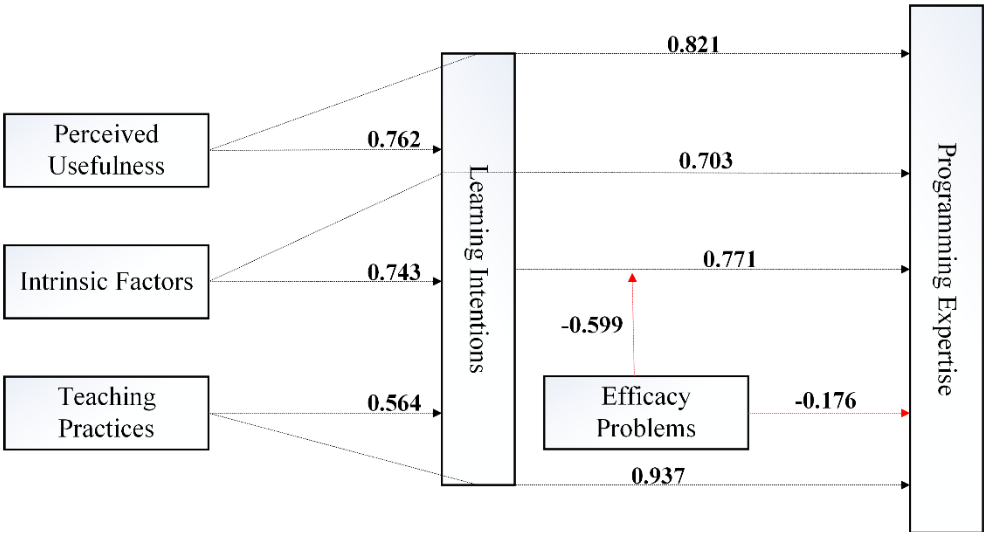

Figure 5. Coefficients of path of the research framework.

Moreover, H7 suggests that LI significantly affects programming expertise, which has also been accepted by the results (β = 0.771, p = 0.000). However, EP hinders the students' ability to learn programming by undermining their potential of learning. The impact of H4 has also been successfully established during the evaluation (β = -0.176, p = 0.017). Increase in EP result in lower programming performance demonstrated by the students. The plausible reason behind this is the students' inability to select relevant information from the internet. Due to the heap of information provided over the internet, deducing timely information is a challenging task. The lack of real-time feedback is a critical problem in programming. Sometimes students find themselves plunged into a programming problem where some support can help them out; however, students lack this support in e-learning. Hence, real-time feedback and support should be provided for effective learning. Recent developments in interactive programming languages and tools can be used to assist learners in writing and compiling their code.

Even students who show high LI may face the issues of lack of real-time support, problems with the complex interface of IDEs, and correct information selection on the internet. In this regard, these factors were supposed to weaken the relationship between LI and programming expertise. Although e-learning provides immense opportunities for the students still, these opportunities may not yield desired results. In this regard, we postulated that students' EP might also moderates the relationship between LI and programming expertise (H9); however, we could not find the significant results for this hypothesis (β = -0.599, p = 0.139). The reasons might embed in the fact that students' LI have a powerful impact on their programming expertise and the impact of EP became insignificant. Considering the evidence that EP adversely influences students' programming expertise, most of the students are successfully achieving programming education and joining the ever-increasing hub of programming experts. The reason behind this is that the students are motivated enough to learn the programming skills and they overcome EP and continue learning programming until they achieve a specific level of expertise whether it takes them more time and energy to assimilate.

H5 implies that PU positively influences LI, which has also been supported by the results (β = 0.762, p = 0.000). Similarly, H8 suggests that the impact of PU on LI trends positively toward programming expertise. This hypothesis has also been successfully accepted during the evaluation (β = 0.821, p = 0.001). This impact corresponds to the fact that students who contemplate that e-learning is beneficial to them are able to perform better in their academic studies and will be able to learn more programming skills.

The above empirical evidence led authors to deduce that in e-learning, IF, TP, and PU are the key factors that serve as an impetus to foster student's LI, which will further contribute in the development of their programming expertise. Mentoring and real-time support will help students to get out of the isolation that students endure during e-learning and will assist them in engaging in programming tasks. By applying this empirical research, human development organizations, government, and the education sector can use web-based e-learning to generate an exceptional pool of talented individuals who can serve effectively to alleviate the poverty and social imbalance. Moreover, they can fulfill the need for human resources in the software sector.

Table 5.

Standardized Coefficients of Structural Model

Hypothesis	Causal Path	β	SE	T-Value	Significance
	Direct Effects				
H1	IF → LI	0.743	0.137	8.353	0.000
H2	TP → LI	0.564	0.141	4.000	0.000
H3	LI → PE	0.771	0.081	9.555	0.000
H4	EP → PE	-0.176	0.074	-2.285	0.017
H5	PU → LI	0.762	0.129	1.456	0.000
H9	LI*EP → PE (Interaction term) student EP negatively related to PE and LI relation with PE weaken the relation	-0.599	0.068	-1.487	0.139
	Indirect Effects				
H6	IF → LI→ PE	0.703	0.171	4.312	0.001
H7	TP→ LI → PE	0.937	0.309	3.036	0.002
H8	PU→ LI → PE	0.821	0.297	0.292	0.001

LI (R^2 = 0.503), PE (R^2 = 0.579)

Conclusion

In this study, we identify and evaluate the factors that influence programming expertise in e-learning. We ascertain teaching practices, intrinsic factors, perceived usefulness, efficacy problems, and learning intentions are the key factors in developing programming skills. A research model has been proposed by extending the technology acceptance model, which integrates all the identified factors. Empirical evidence indicates that effective teaching practices, perceived usefulness, and correct intrinsic motivations are the bases to instigate the aspiration to learn programming. Students' efficacy problems undermine their ability to learn; however, they do not impact their programming skills significantly. At the institutional level, effective learning management systems should be provided that may encompass the features of face-to-face communication in e-learning. Moreover, effective student-teacher interaction needs to be established as students need immediate help during the programming problems. The availability of quick response can be highly effective as the students sometimes plunge into problems and lose motivation. Specifically, when students confront complex programming tasks, they need spontaneous help to grasp their motivation for learning and to complete the programming tasks.

Step-by-step programming tutorials with the formal lectures can also help students to grasp a basic understanding of the programming activity discussed during the lectures.

Furthermore, the mentoring and support staff can help students in completing their complex programming tasks. It will benefit the student to come out of the problem of isolation and break the self-centered view of learning which will result in broadening their knowledge horizons. Moreover, students' support will be beneficial to overcome the stress of learning everything by themselves. Therefore, an interactive teaching environment and immediate assistance can help students in enhancing learning intentions of the students, which will overcome the efficacy problems and facilitate them in attaining right programming expertise in the web-based e-learning environment.

Discussion

The experimental analysis presented in the results section demonstrates that TP plays a pivotal role in instigating learning motivations for programming. These findings have been aligned with the research of Dos Santos and Cechinel (2018), which supports that effective teaching style yields better learning outcomes in students. Effective TP involves more interaction between students and teachers, which leads students to grasp the contents of the lectures adequately. The interactivity of the e-learning system can help in solving the face-to-face learning issue in the e-learning paradigm. This outcome is associated with the findings of Pituch and Lee (2006) and Chen (2011) who reveal that the interactivity of the LMS and immediate response motivates the learner in e-learning.

We propose that IF positively affects LI where the essential elements in the IF include the student's perception and motivation toward learning. The empirical results of our study claim that these factors have a positive impact on the student's LI. This finding is in line with the results of Pugh (2019) who argues that the student's motivation is the key success factor in higher education. This impact has also been demonstrated by Venkatesh (2000) who states that IF positively impacts the learner's motivation. However, in e-learning, it is difficult to have interactive sessions with the mentor; thus, LMS should be developed in a way that it can provide a platform for face-to-face discussions with the tutors along with the text discussions. In this regard, our study proves that IF also positively affect the LI of students toward programming.

This study demonstrates that PU plays an important role in developing LI that further contributes to programming expertise. The items in PU correspond to the fact that career-oriented professionals grasp more knowledge in e-learning. This hypothesis is consistent with Nganji (2018) and Wingo et al. (2017) who argue that career-oriented learners use strategic methods to complete their tasks in time and perform well during examination and thus are able to get the valuable learning outcomes. Similarly, the impact of PU is also aligned with the Park (2009) and Van Raaij and Schepers (2008) who demonstrate that the success in e-learning is dependent on the usefulness of the e-learning system.

The result of H4 suggested that EP negatively affect the programming expertise. Here, efficacy is concerned with the contextual problems faced by the student during programming including difficulty in using IDEs, getting online help, and lack of support. Although numerous interactive IDEs have been developed, however, students still face problems while using them. The research of Altınay (2017) demonstrates the effectiveness of peer learning that can improve the online learning process. Peer learning uses the experience of other students, their motivation, and social interaction to help the other students in e-learning. In this regard, the feedback and support for e-learning students are necessary; Tsai (2013) conducted an empirical analysis of e-learning students and demonstrate that students who receive immediate feedback on their learning process perform better than other e-learning students. Tang, Tang, and Chiang (2014) demonstrate the positive impact of learning from the online resources; moreover, they demonstrate that students continue visiting a website/blog if they get required help from it. In addition to student-teacher communication, student-student interaction should also be provided because students can communicate easily with their peers as compared to their teachers. Providing students with step-by-step solutions for the programming tasks will increase their interest in programming tasks. Online advising and mentoring services can also help e-learning students to discuss their problems. Finally, the students need to be satisfied enough about their study program before joining a course in the e-learning. Moreover, interactive LMS and responsive teaching facilities should be provided to the students, which can highly contribute to the programming skills development in e-learning.

Web technologies have been providing immense opportunities for students worldwide to learn state-of-the-art courses using e-learning. For effective programming education, the practitioners should provide more support to the students using LMS that may include video conferencing services for real-time student-teacher interaction.

Acknowledgements

This research is supported by the National Key Research and Development Program of China (no. 2017YFB1001801), the National Science Foundation of China under grant no. 61672276, and the Collaborative Innovation Centre of Novel Software Technology and Industrialization, Nanjing University.

Author Notes

The corresponding author is Wanchun Dou, Department of Computer Science and Technology, Nanjing University, P. R. China and State Key Laboratory of Novel Software Technology, Nanjing University, P. R. China, email: douwc@nju.edu.cn

References

Alharbi, S., & Drew, S. (2014). Using the technology acceptance model in understanding academics' behavioural intention to use learning management systems. *International Journal of Advanced Computer Science and Applications, 5*(1), 143-155.

Allen, E., Cartwright, R., & Stoler, B. (2002). DrJava: a lightweight pedagogic environment for Java. In *Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education* (pp. 137-141). Cincinnati, Ohio: ACM.

Altınay, Z. (2017). Evaluating peer learning and assessment in online collaborative learning environments. *Behaviour & Information Technology, 36*(3), 312-320. doi: 10.1080/0144929X.2016.1232752.

Amemiya, Y., & Anderson, T. W. (1990). Asymptotic chi-square tests for a large class of factor analysis models. *The Annals of Statistics*, *18*(3), 1453-1463.

Ateş Çobanoğlu, A. (2018). Student teachers' satisfaction for blended learning via Edmodo learning management system. *Behaviour & Information Technology, 37*(2), 133-144. doi: 10.1080/0144929X.2017.1417481.

Berry, S. (2019). Teaching to connect: Community-building strategies for the virtual classroom. *Online Learning, 23*(1), 164-183. doi: 10.24059/olj.v23i1.1425.

Boelens, R., De Wever, B., & Voet, M. (2017). Four key challenges to the design of blended learning: A systematic literature review. *Educational Research Review, 22*, 1-18.

Bouhnik, D., & Marcus, T. (2006). Interaction in distance-learning courses. *Journal of the Association for Information Science and Technology, 57*(3), 299-305.

Celepkolu, M., & Boyer, K. E. (2018). The importance of producing shared code through pair programming. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 765-770). Baltimore, Maryland: ACM.

Chen, J.-L. (2011). The effects of education compatibility and technological expectancy on e-learning acceptance. *Computers & Education, 57*(2), 1501-1511.

Cheng, Y. M. (2011). Antecedents and consequences of e-learning acceptance. *Information Systems Journal, 21*(3), 269-299.

Chin, W. W. (1998). The partial least squares approach to structural equation modeling. *Modern Methods for Business Research, 295*(2), 295-336.

Dahalan, N., Hassan, H., & Atan, H. (2012). Student engagement in online learning: Learners attitude toward e-mentoring. *Procedia-Social and Behavioral Sciences, 67*, 464-475.

Davis, F. D., & Venkatesh, V. (1996). A critical assessment of potential measurement biases in the technology acceptance model: three experiments. *International Journal of Human-computer Studies, 45*(1), 19-45.

Dorner, H., & Kárpáti, A. (2010). Mentoring for innovation: Key factors affecting participant satisfaction in the process of collaborative knowledge construction in teacher training. *Online Learning, 14*(4), 63-77. doi: 10.24059/olj.v14i4.127.

dos Santos, H. L., & Cechinel, C. (2018). The final year project supervision in online distance learning: assessing students and faculty perceptions about communication tools. *Behaviour & Information Technology*, *38,* 1-20. doi: 10.1080/0144929X.2018.1514423.

Eom, S. B., Wen, H. J., & Ashill, N. (2006). The determinants of students' perceived learning outcomes and satisfaction in university online education: An empirical investigation. *Decision Sciences Journal of Innovative Education, 4*(2), 215-235.

Galusha, J. M. (1998). Barriers to learning in distance education, *A publication of University of Southern Mississippi, USA.* ED416377, 1-23.

Hair, J., Black, W., BABIN, B. Y. A., Anderson, R., & Tatham, R. (2010). *Multivariate data analysis: A global perspective*. London: Pearson Prentice Hall.

Hendijani, R., Bischak, D. P., Arvai, J., & Dugar, S. (2016). Intrinsic motivation, external reward, and their effect on overall motivation and performance. *Human Performance, 29*(4), 251-274. doi: 10.1080/08959285.2016.1157595.

Hu, L., & Bentler, P. M. (1999). Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural Equation Modeling: A Multidisciplinary Journal, 6*(1), 1-55.

Hung, M.-L., & Chou, C. (2015). Students' perceptions of instructors' roles in blended and online learning environments: A comparative study. *Computers & Education, 81*, 315-325.

Jan, A. U., & Contreras, V. (2011). Technology acceptance model for the use of information technology in universities. *Computers in Human Behavior, 27*(2), 845-851.

Jenkins, T. (2001). The motivation of students of programming. In *Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education* (pp. 53-56). Canterbury, United Kingdom, ACM.

Jose, G. S. S., & Christopher, C. S. (2018). Secure cloud data storage approach in e-learning systems. *Cluster Computing*, 1-6.

Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR), 37*(2), 83-137.

Khan, A. S., & Nawaz, A. (2013). Role of contextual factors in using e-learning systems for higher education in developing countries. *Journal of Educational Research and Studies, 1*(4), 27-34.

Lam, M., Chan, E., Lee, V., & Yu, Y. (2008). Designing an automatic debugging assistant for improving the learning of computer programming. In *Proceedings of the 1st International Conference on Hybrid Learning and Education* (pp. 359-370). Hong Kong, China.

Law, K. M., Lee, V. C., & Yu, Y.-T. (2010). Learning motivation in e-learning facilitated computer programming courses. *Computers & Education, 55*(1), 218-228.

Lee, B.-C., Yoon, J.-O., & Lee, I. (2009). Learners' acceptance of e-learning in South Korea: Theories and results. *Computers & Education, 53*(4), 1320-1329.

Lee, Y.-H., Hsiao, C., & Purnomo, S. H. (2014). An empirical examination of individual and system characteristics on enhancing e-learning acceptance. *Australasian Journal of Educational Technology, 30*(5), 562-579.

Liaw, S.-S. (2008). Investigating students' perceived satisfaction, behavioral intention, and effectiveness of e-learning: A case study of the Blackboard system. *Computers & Education, 51*(2), 864-873.

Linnenbrink, E. A., & Pintrich, P. R. (2002). Motivation as an enabler for academic success. *School Psychology Review, 31*(3), 313-327.

Martín-Rodríguez, Ó., Fernández-Molina, J. C., Montero-Alonso, M. Á., & González-Gómez, F. (2015). The main components of satisfaction with e-learning. *Technology, Pedagogy and Education, 24*(2), 267-277. doi: 10.1080/1475939X.2014.888370.

Martin, F., Budhrani, K., Kumar, S., & Ritzhaupt, A. (2019). Award-winning faculty online teaching practices: Roles and competencies. *Online Learning, 23*(1), 184-205. doi: 10.24059/olj.v23i1.1329.

Nganji, J. T. (2018). Towards learner-constructed e-learning environments for effective personal learning experiences. *Behaviour & Information Technology, 37*(7), 647-657. doi: 10.1080/0144929X.2018.1470673.

Offir, B., Lev, Y., & Bezalel, R. (2008). Surface and deep learning processes in distance education: Synchronous versus asynchronous systems. *Computers & Education, 51*(3), 1172-1183.

Park, S. Y. (2009). An analysis of the technology acceptance model in understanding university students' behavioral intention to use e-learning. *Educational Technology & Society, 12*(3), 150-162.

Pituch, K. A., & Lee, Y.-k. (2006). The influence of system characteristics on e-learning use. *Computers & Education, 47*(2), 222-244.

Pugh, C. (2019). Self-determination: Motivational profiles of bachelor's degree seeking students at an online, for-profit university. *Online Learning, 23*(1), 111-131. doi: 10.24059/olj.v23i1.1422.

Raigoza, J. (2017). *A study of students' progress through introductory computer science programming courses.* Paper presented at the 2017 IEEE Frontiers in Education Conference (FIE), (pp. 1-7). Indianpolis, Indiana: IEEE Education Society.

Sarpong, K. A.-M., Arthur, J. K., & Amoako, P. Y. O. (2013). Causes of failure of students in computer programming courses: The teacher-learner perspective. *International Journal of Computer Applications, 77*(12), 27-32.

Sekaran, U., & Bougie, R. (2011). *Business research methods: A skill-building approach*. New York: McGraw-Hill.

Tang, J.-t. E., Tang, T.-I., & Chiang, C.-H. (2014). Blog learning: effects of users' usefulness and efficiency toward continuance intention. *Behaviour & Information Technology, 33*(1), 36-50. doi: 10.1080/0144929X.2012.687772.

Tarhini, A., Hone, K., & Liu, X. (2014). The effects of individual differences on e-learning users' behaviour in developing countries: A structural equation model. *Computers in Human Behavior, 41*, 153-163.

Tenenhaus, M., Vinzi, V. E., Chatelin, Y.-M., & Lauro, C. (2005). PLS path modeling. *Computational Statistics & Data Analysis, 48*(1), 159-205.

Tsai, C.-W. (2013). An effective online teaching method: The combination of collaborative learning with initiation and self-regulation learning with feedback. *Behaviour & Information Technology, 32*(7), 712-723. doi: 10.1080/0144929X.2012.667441.

Valentine, D. (2002). Distance learning: Promises, problems, and possibilities. *Online Journal of Distance Learning Administration 5(3),* 1-11.

Van Raaij, E. M., & Schepers, J. J. (2008). The acceptance and use of a virtual learning environment in China. *Computers & Education, 50*(3), 838-852.

Venkatesh, V. (2000). Determinants of perceived ease of use: Integrating control, intrinsic motivation, and emotion into the technology acceptance model. *Information Systems Research, 11*(4), 342-365.

Venkatesh, V., & Davis, F. D. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management Science, 46*(2), 186-204.

Wingo, N. P., Ivankova, N. V., & Moss, J. A. (2017). Faculty perceptions about teaching online: Exploring the literature using the technology acceptance model as an organizing framework. *Online Learning, 21*(1), 15-35.

Xia, B. S., & Liitiäinen, E. (2017). Student performance in computing education: An empirical analysis of online learning in programming education environments. *European Journal of Engineering Education, 42*(6), 1025-1037. doi: 10.1080/03043797.2016.1250066.

Yu, Y., Poon, C., & Choy, M. (2006). *Experiences with PASS: Developing and using a programming assignment assessment system.* Paper presented at the Sixth International Conference on Quality Software. Beijing: IEEE.

Yunkul, E., & Cankaya, S. (2017). Students' attitudes towards Edmodo, a social learning network: A scale development study. *Turkish Online Journal of Distance Education, 18*(2), 16-29.

Zin, A. M., Idris, S., & Subramaniam, N. K. (2006). Improving learning of programming through e-learning by using asynchronous virtual pair programming. *Turkish Online Journal of Distance Education, 7*(3), 162-173.