Research Article
American Journal of Science Education Research

What Factors Most Influence Self-Learning Python Programming For 10th Grade Students? A Case Study in Viet Nam

Hung Van Tran[1*], Toan Nguyen Dinh[2], Tu Le Dang Anh[3], Vuong Ho Quoc[4]

[1,2,3,4]University of DaNang, University of Science and Education, DaNang, VietNam.

[*]**Corresponding author:** Hung Van Tran, University of DaNang, University of Science and Education, DaNang, Vietnam. Email: tvhung@ued.udn.vn

Citation: Tran VH, Dinh TN, Anh TLD, Quoc VH (2024) What Factors Most Influence Self-Learning Python Programming For 10th Grade Students? A Case Study in Viet Nam. American J Sci Edu Re: AJSER-193.

Received Date: June 21, 2024; **Accepted Date:** June 26, 2024; **Published Date:** July 02, 2024

Abstract

Self-learning is one of the cores and general abilities emphasized by the Ministry of Education and Training the 2018 general education curriculum. Coupled with the development of modern technology, Python is becoming popular and is a trend in modern technology and education. This study aims to assess the current status of Grade 10 students' self-learning in Python programming at high schools DaNang city and propose solutions to improve the effectiveness of self-learning in programming. In this research, a validated survey tool was used to collect information from 620 Grade 10 students from high schools in Da Nang city. The collected data were then analysed to evaluate the current state of self-learning programming among Grade 10 students. From the analysis results, the study proposes several solutions to improve the effectiveness and quality of students' self-learning programming, such as suggesting more effective learning methods, providing necessary support resources, and giving guidance to students. These solutions also serve as materials to help students learn how to develop programming skills and achieve success in self-learning.

Keywords: *Self-learning; programming skills; Grade 10; Python.*

Introduction

In the context of the 21st century, as information technology increasingly plays a central role in life and work, self-learning programming has become not just a trend but an essential requirement. The development of online educational platforms and open learning resources has made this easier than ever, allowing individuals of all ages and professional backgrounds the opportunity to develop programming skills. The flexibility and accessibility of online learning have encouraged many to equip themselves with this skill, leading to significant changes in how we approach education and career development. In its 2018 General Education Program, the Vietnam Ministry of Education and Training emphasized the development of self-learning capabilities, prioritizing the encouragement of students to learn programming not just as a need but as a top priority. Studies [1,2] have shown that students' involvement in self-learning programming not only helps them learn technical skills but also develops problem-solving abilities and critical thinking.

Other studies have also emphasized the importance of enhancing self-learning programming techniques in general education, highlighting the necessity of providing students with effective strategies and resources to support self-learning in programming [3]. In a comparative study on self-learning programming methods, Chusni et al. (2021) [4] stressed the importance of considering different educational contexts, revealing the varying effectiveness of methods depending on factors such as the school environment and teaching approach. Lastly, a study by A Stocco (2019) [5] identified challenges and opportunities in encouraging self-learning programming among high school students, shedding light on potential barriers students may face and proposing strategies for teachers to create a conducive learning environment.

The purposes of this study are to explore factors affecting self-learning in programming among 10th-grade students in high schools DaNang, Vietnam, and to propose specific solutions and implementation methods to support students in self-learning Python programming.

Literature

Python, due to its simplicity and flexibility, has become one of the leading programming languages widely used in education. With its easy-to-understand syntax and extensive libraries, Python supports not only basic programming learning but also complex applications such as web development, data science, and artificial intelligence [5]. Teaching Python in schools helps students develop problem-solving skills, logical thinking, and computer science reasoning, crucial foundations in the digital age [6]. Self-learning programming methods include using online platforms like Codecademy and Khan Academy, offering video lectures, hands-on exercises, and real-world projects. Studies have shown that a highly interactive learning environment with community support can significantly improve learning outcomes [7].

According to the theory of self-directed learning, it is crucial for students to actively set goals, track progress, and evaluate their own results. Motivation to learn, especially in programming, is often driven by an understanding of the purpose of learning and expectations of meaningful and practical learning outcomes [8].

Jingming (2007) [9] evaluated programming education in general education and noted that, despite an increase in the number of courses, the quality and approach to education still vary. This calls for greater attention to curriculum design, particularly in diversifying content and teaching methods to meet the varied needs of students.

The current trend in self-learning programming is the combination of online learning and face-to-face education, providing a blended learning environment that allows students to control their learning pace. New online tools and resources have also been developed to help students learn anytime, anywhere, with technical support when needed [10]. Previous initiatives like the "Hour of Code" program have demonstrated potential in introducing programming to students early. However, for long-term effectiveness, these programs need to be systematically integrated into the regular curriculum, with periodic evaluations and monitoring to ensure that students not only learn to code but also develop critical thinking and problem-solving skills [11].

Proposed Model and Research Hypotheses

Technology Acceptance Model (TAM): The Technology Acceptance Model (TAM), designed to assess user attitudes towards information systems and predict their acceptance [12], focuses on two main factors: Perceived Usefulness (the degree to which a user believes that using a specific system will enhance their job performance) and Perceived Ease of Use (the degree to which a user believes that they can use the system without effort). The combination of these perceptions contributes to the intention to use technology, leading to actual acceptance behavior. The TAM model has proven to be an effective analytical tool in predicting and explaining technology acceptance in various contexts. (See Figure 1: Technology Acceptance Model TAM [12].

Unified Theory of Acceptance and Use of Technology (UTAUT): The UTAUT model is an advancement from TAM [13], synthesizing eight previous theoretical models to explain the behavioral intention and usage behavior of users towards new information systems. This model focuses on four main factors: Performance Expectancy, Effort Expectancy, Social Influence, and Facilitating Conditions. Performance Expectancy is defined as the degree to which an individual believes that using the new system will help them achieve effectiveness in work or learning; Effort Expectancy is defined as the ease or difficulty of using the system; Social Influence is seen as the degree to which an individual feels pressured by the idea of others around them that they should use the new system; Facilitating Conditions are defined as the degree to which an individual believes that the technical infrastructure of the organization is adequate to support users to adopt the system more conveniently.

Choosing the UTAUT model (Unified Theory of Acceptance and Use of Technology) is a rational and effective choice for researching factors affecting the self-learning of Python programming among 10th-grade students. The UTAUT model provides a comprehensive theoretical framework to understand the factors influencing the acceptance and use of technology, especially in the educational context. This model allows us to integrate various aspects affecting the self-learning programming process, from personal perceptions about benefits and technology accessibility to influences from peers and family, as well as support from schools and educational tools. This study will utilize the UTAUT model to further assess how these factors interact and influence the intention and behavior of self-learning Python programming, thereby contributing to the development of more effective educational strategies to promote initiative and autonomy in students' learning.

Proposed Research Model

Focusing on analyzing the current situation and solutions for self-learning Python programming for 10th-grade students in Da Nang city, the author proposes the following research model. It includes:

- Perceived Usefulness (PU) in the study is understood as recognizing the usefulness of learning Python programming. This factor includes 4 observations: Perception that learning Python programming improves problem-solving skills (PU1); Belief that Python skills will be useful for learning other subjects (PU2); Expectation that knowing Python programming will open up future career opportunities (PU3); Perception that Python programming helps in effective independent learning (PU4).
- Perceived Ease of Use (PEU) is recognizing that Python is easy to learn and use. It includes 4 observations: Perception that Python language has simple and easy-to-understand syntax (PEU1); Existence of many free and online learning resources (PEU2); Awareness of the ease of finding help from the Python programming community (PEU3); Feeling that the Python development environment (e.g., IDLE, PyCharm) is user-friendly for beginners (PEU4).
- Social Influence (SI) focuses on the relationships around students influencing their learning. This factor includes 4 observations: Support from friends (SI1); Influence of teachers (SI2); Role models from siblings or relatives (SI3); Active participation in the community (SI4).
- Facilitating Conditions (FC) reflected in 4 observations: Availability of computers and necessary software at home or school (FC1); Access to stable internet for online programming learning (FC2); Organization of Python classes and workshops at school (FC3); Support from programming communities (FC4).

Research Hypotheses

Based on studies on the TAM, UTAUT models, and the proposed research model, the author proposes the following hypotheses: H1: Perceived Usefulness directly influences the intention to self-learn Python programming; H2: Perceived Ease of Use directly influences the intention to self-learn Python programming; H3: Social Influence directly influences the intention to self-learn Python programming; H4: Facilitating Conditions directly influence the behavior of self-learning Python programming.

Method

Reliability Testing of the Scale

A survey method was used to collect primary data related to the status of self-learning Python programming among 10th graders in Da Nang city. The total sample size for the survey was 620 students. A 5-point Likert scale (from 1-5 levels) was used in the questions. Survey data were processed using SPSS 24.0 for descriptive statistics and basic statistical hypothesis testing. To assess the research model and the relationships between variables, we used Exploratory Factor Analysis (EFA) and Confirmatory Factor Analysis (CFA) to determine the structure of observed and latent variables. Next, the Structural Equation Modeling (SEM) was analyzed through SmartPLS software, allowing us to evaluate the reliability and effectiveness of the research model based on the PLS-SEM method. This result not only provides information about the model's fit to the collected data but also helps clarify the relationships with factors influencing self-learning Python programming among 10th graders in Da Nang.

Table 1 presents the Cronbach's Alpha reliability results for each factor measured in the study. The Cronbach's Alpha values indicate high reliability for each scale: Perceived Usefulness (0.822), Perceived Ease of Use (0.831), Social Influence (0.861), and Facilitating Conditions (0.842). According to Hair et al. (2010) [14], a Cronbach's Alpha value greater than 0.7 is considered acceptable to affirm the homogeneity and reliability of the scale. These results indicate that the scales used in the study are capable of measuring the factors influencing the self-learning of Python programming among 10th graders accurately and consistently. The high homogeneity of the items within each scale also reflects a good consensus in the respondents' understanding and evaluation of the study factors.

Research Results

Table 1: Scale Reliability.

Factors	Cronbach's Alpha
Perceived Usefulness	0,822
Perceived Ease of Use	0,831
Social Influence	0,861
Facilitating Conditions	0,842

Exploratory Factor Analysis (EFA) Results

- Factor Impact Analysis: This study explores the factors affecting the self-learning of Python programming among 10th-grade students in Da Nang, analyzing data from 620 students. To ensure data suitability for factor analysis, we used the Kaiser-Meyer-Olkin (KMO) measure of sampling adequacy, which is quite high (.801), supporting the suitability of the factor analysis. Bartlett's test of sphericity was significant (χ^2 = 5896.740, $p < .001$), indicating that the variables are sufficiently related to each other to perform factor analysis, meeting the standards set by Kaiser (1974) [15]. The high KMO value suggests that the observed variables in the dataset are significantly correlated, suitable for conducting Exploratory Factor Analysis (EFA). Bartlett's Test of Sphericity results show a chi-square value of 5896.740 with degrees of freedom (df) of 231 and a Sig. value of 0.000. This confirms that the variables in the dataset are strongly correlated enough to fit factor analysis. The low p-value (below 0.05) from Bartlett's test confirms that the correlation matrix is not an identity matrix and that the variables are significantly correlated with each other, strongly supporting factor analysis.

Table 2: KMO and Bartlett's Test

Kaiser-Meyer-Olkin Measure of Sampling Adequacy.		.801
Bartlett's Test of Sphericity	Approx. Chi-Square	5896.740
	df	231
	Sig.	.000

The high KMO value and Bartlett's test results confirm the suitability of the data for factor analysis. This indicates that factor analysis is an appropriate method to continue exploring the factor structure of the observed variables, helping to identify the main underlying factors behind these variables. The next steps include performing Exploratory Factor Analysis to determine the number and nature of factors, using rotation methods (such as Varimax) to clarify the relationships between variables and each factor, and validating the reliability of the factors by calculating Cronbach's Alpha for each extracted factor.

Table 3: Rotated Factor Matrix.

	Rotated Factor Matrix[a] Component					
	1	2	3	4	5	6
XH2	.828					
XH4	.801					
XH3	.749					
XH1	.695					
DK3		.788				
DK2		.755				
DK1		.724				
DK4		.552				
DSD3			.791			
DSD2			.763			
DSD4			.736			
DSD1			.636			
HI4				.743		
HI1				.723		
HI2				.632		
HI3				.586		
TH3					.785	
TH1					.750	
TH2					.665	
YDTH2						.763
YDTH3						.734
YDTH1						

Multivariate Linear Regression Analysis Results

To examine the differences in self-learning levels among student groups with varying support from family, environment, and society, we applied ANOVA. The results showed significant differences in self-learning scores based on these groups ($p < 0.05$), supporting the research by Fisher (1925) [16] on the application of ANOVA in educational research.

Table 4: ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	89.372	5	17.874	26.926	.000[b]
	Residual	407.597	614	.664		
	Total	496.970	619			
a. Dependent Variable: X_YDTH						
b. Predictors: (Constant), X_DSD, X_XH, X_DK, X_HI						

These results indicate that external factors such as family support, educational environment, and societal influence play significant roles in the self-learning abilities of students. Identifying these differences allows for more targeted educational strategies and interventions to enhance self-learning outcomes among students, especially in programming education where individual pace and preferences can vary greatly. Further research could delve into specific interventions or educational practices that could mitigate these differences and promote more equitable self-learning opportunities across different student demographics.

Table 5: Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Durbin-Watson
1	.424[a]	.180	.173	.81476	1.626
a. Predictors: (Constant), X_SD, X_XH, X_DK, X_HI					
b. Dependent Variable: X_YDTH					

Table 6: Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients			Collinearity Statistics
		B	Std. Error	Beta	t	Sig.	Tolerance
1	(Constant)	1.737	.247		7.045	.000	
	X_HI	.167	.051	.143	3.255	.001	.694
	X_DSD	-.072	.035	.080	-2.044	.041	.879
	X_XH	-.044	.033	.051	-1.326	.185	.892
	X_DK	.081	.043	.079	1.894	.049	.775
a. Dependent Variable: X_YDTH							

Confirmatory Factor Analysis (CFA) Results

In this study, Confirmatory Factor Analysis (CFA) was applied to validate the factor structure of the observed variables influencing the self-learning of Python programming among 10th-grade students. The results indicated that factors such as perceived usefulness, ease of use, social influence, and facilitating conditions all have high standardized regression weights and are statistically significant, demonstrating their significant impact and stability within the research model. Basic indices such as Chi-square/df = 1.843 (≤ 3) indicate good model fit, CFI = 0.924 suggests good compatibility with the data; GFI = 0.803, although just acceptable, still meets the minimum requirement of 0.8; TLI = 0.930 emphasizes the model's appropriateness, and RMSEA = 0.069 indicates a good fit of the model with the collected data. These indices confirm that the CFA model fits well with the initially declared factor structure.

Table 7: Regression Weights.

			Estimate	S.E.	C.R.	P
DK2	<---	DK	1.000			
DK4	<---	DK	.945	.043	21.839	.000
DK3	<---	DK	.870	.042	20.769	.000
DK1	<---	DK	.739	.041	18.194	.000
XH3	<---	XH	1.000			
XH2	<---	XH	.969	.051	19.046	.000
XH1	<---	XH	.798	.043	18.591	.000
XH4	<---	XH	.620	.043	14.531	.000
DSD3	<---	DSD	1.000			
DSD2	<---	DSD	1.071	.057	18.917	.000
DSD4	<---	DSD	1.054	.056	18.847	.000
DSD1	<---	DSD	.735	.049	15.105	.000
HI4	<---	HI	1.000			
HI1	<---	HI	.968	.050	19.189	.000
HI2	<---	HI	.830	.049	16.890	.000
HI3	<---	HI	.796	.049	16.325	.000
TH3	<---	TH	1.000			
TH1	<---	TH	1.006	.058	17.368	.000
TH2	<---	TH	.811	.050	16.202	.000
YDTH2	<---	YDTH	1.000			
YDTH3	<---	YDTH	.933	.070	13.370	.000
YDTH1	<---	YDTH	.650	.059	11.113	.000

The Standardized Regression Weights further evaluated to determine the explanatory power of the observed variables on the parent factors. Results shown in Table 8 indicate that all observed variables have standardized impact coefficients in the Estimate column greater than 0.5, and even above 0.7, indicating that these observed variables effectively explain the parent factors.

Table 8: Standardized Regression Weights.

			Estimate
DK2	<---	DK	.850
DK4	<---	DK	.804
DK3	<---	DK	.769
DK1	<---	DK	.692
XH3	<---	XH	.791
XH2	<---	XH	.789
XH1	<---	XH	.767
XH4	<---	XH	.607
DSD3	<---	DSD	.808
DSD2	<---	DSD	.773
DSD4	<---	DSD	.770
DSD1	<---	DSD	.624
HI4	<---	HI	.824
HI1	<---	HI	.769

HI2	<---	HI	.681
HI3	<---	HI	.661
TH3	<---	TH	.786
TH1	<---	TH	.805
TH2	<---	TH	.708
YDTH2	<---	YDTH	.797
YDTH3	<---	YDTH	.749
YDTH1	<---	YDTH	.527

The Cronbach's Alpha (CR) indices demonstrate high reliability in the model's factors, with TH, YDTH, HI, DSD, DK, and XH all having CR values above 0.7. This reflects the homogeneity and reliability of the observed variables within each group, providing an important basis for validating the factors' validity in the study.

In terms of convergent validity, measured by the Average Variance Extracted (AVE), factors TH, YDTH, HI, and XH all surpass the acceptance threshold of 0.5, indicating that a substantial amount of the variance of the observed variables is explained by these factors. However, DSD with an AVE of only 0.371 does not meet this threshold, suggesting this variable may not effectively explain the variance of the observed variables, requiring improvements or reconsideration.

Discriminant validity, assessed through MSV and MaxR(H), shows that most factors have MaxR(H) higher than their correlations with other factors, particularly for factors TH, YDTH, HI, DK, and XH. This confirms that these factors are relatively independent from each other, supporting good discriminant validity in the model. Conversely, DSD shows deficiencies once again with a lower MaxR(H) than its correlations with another factor, indicating potential redundancy that could affect the accuracy of the model.

In conclusion, the current research model shows good reliability, convergent, and discriminant validity for most factors, except for the DSD factor, which requires improvement. Refining or adding more observed variables for DSD could enhance this factor's explanatory capability and independence within the overall model.

Table 9: AVE, MSV, and Fornell and Larcker table.

	CR	**AVE**	**MSV**	**MaxR**	**TH**	**YDTH**	**HI**	**DSD**	**DK**	**XH**
TH	0,81	0.85	0.156	0.850	**0.850**					
YDTH	0,727	0.804	0.156	0.804	0.395	**0.804**				
HI	0,822	0.802	0.217	0.802	0.206	0.162	**0.802**			
DSD	0,831	0.371	0.217	0.371	0.223	0.174	0.140	**0.371**		
DK	0,842	0.779	0.055	0.779	-0.064	0.018	0.235	-0.286	**0.779**	
XH	0,861	0.806	0.217	0.806	0.177	0.151	0.466	0.201	0.129	**0.806**

(Note: Actual tables were mentioned for presentation and are to be included as detailed in the research data or results section.)

Structural Equation Modeling (SEM) Results

In this model, several significant relationships were highlighted, including between YDTH and variables XH and YDTH itself, where the estimated regression coefficient for YDTH influenced by XH is 0.135 with a high level of significance (p-value < 0.001). Similarly, the relationship between YDTH and itself shows a coefficient of 1.006, indicating a strong self-interaction through different cycles. This relationship suggests that YDTH is not only influenced by XH but also has a strong interaction with itself. The variable HI also shows a significant relationship with YDTH1 with a regression coefficient of 0.656, indicating a positive influence of HI on YDTH1. This may indicate that support or impact from HI significantly affects YDTH1 within the model. Variables in DSD also show significant effects on other variables, with all DSD variables (DSD1, DSD2, DSD3, DSD4) having regression coefficients ranging from 0.732 to 1.068, all statistically significant with $p < 0.001$. This highlights the clear impact of factors related to DSD on other variables in the model. Similarly, variables XH and DK also show strong relationships with other variables in the model with high coefficients and reliability. These results provide insights into how variables interact within the model, thereby better understanding the factors affecting YDTH and other related variables.

Table 10: Regression Weights

			Estimate	S.E.	C.R.	P
YDTH	<---	XH	.135	.022	-5.994	.000
YDTH	<---	DSD	.189	.024	-7.775	.000
YDTH	<---	HI	1.073	.105	10.198	.000
TH	<---	YDTH	1.560	.120	13.044	.000
TH	<---	DK	.115	.026	-4.385	.000
DK2	<---	DK	1.000			
DK4	<---	DK	.953	.044	21.646	.000
DK3	<---	DK	.870	.043	20.410	.000
DK1	<---	DK	.749	.041	18.234	.000

XH3	<---	XH	1.000			
XH2	<---	XH	.967	.051	18.805	.000
XH1	<---	XH	.783	.043	18.163	.000
XH4	<---	XH	.604	.043	14.154	.000
DSD3	<---	DSD	1.000			
DSD2	<---	DSD	1.068	.057	18.662	.000
DSD4	<---	DSD	1.050	.057	18.574	.000
DSD1	<---	DSD	.732	.049	14.935	.000
HI4	<---	HI	1.000			
HI1	<---	HI	1.048	.106	9.877	.000
HI2	<---	HI	1.003	.102	9.801	.000
HI3	<---	HI	.852	.097	8.818	.000
YDTH2	<---	YDTH	1.000			
YDTH3	<---	YDTH	1.006	.092	10.897	.000
YDTH1	<---	YDTH	.656	.081	8.064	.000

Moving on to the Standardized Regression Weights, this table presents the normalized regression coefficients. We will rely on the regression coefficient Estimate in this table to assess the impact level of independent variables on the dependent variable. According to Table 11, among the three variables affecting YDTH, the order of impact decreases as follows: HI, DSD, XH.

Table 11: Standardized Regression Weights.

			Estimate
YDTH	<---	XH	.421
YDTH	<---	DSD	.610
YDTH	<---	HI	.925
TH	<---	YDTH	.990
TH	<---	DK	.539
DK2	<---	DK	.846
DK4	<---	DK	.807
DK3	<---	DK	.765
DK1	<---	DK	.697
XH3	<---	XH	.797
XH2	<---	XH	.793
XH1	<---	XH	.758
XH4	<---	XH	.595
DSD3	<---	DSD	.806
DSD2	<---	DSD	.770
DSD4	<---	DSD	.766
DSD1	<---	DSD	.621
HI4	<---	HI	.526
HI1	<---	HI	.531
HI2	<---	HI	.525
HI3	<---	HI	.451
TH3	<---	TH	.747
TH1	<---	TH	.774
TH2	<---	TH	.733
YDTH2	<---	YDTH	.555
YDTH3	<---	YDTH	.561
YDTH1	<---	YDTH	.380

Finally, we consider the Squared Multiple Correlations table. This table represents the R-squared values, which measure the impact level of independent variables on the dependent variable. According to Table 12, the R-squared value for YDTH is 0.563 or 56.3%, indicating that the independent variables account for 56.3% of the variance in YDTH.

Table 12: Squared Multiple Correlations

	Estimate
YDTH	.563
YDTH1	.545
YDTH3	.415
YDTH2	.608
TH2	.537
TH1	.599
TH3	.558
HI3	.604
HI2	.875
HI1	.682
HI4	.776
DSD1	.585
DSD4	.586
DSD2	.593
DSD3	.650
XH4	.454
XH1	.575
XH2	.629
XH3	.635
DK1	.486
DK3	.585
DK4	.651
DK2	.715

Proposal

These tables and results from the SEM analysis offer a comprehensive view of the model's dynamics, showing significant dependencies and interactions among variables. This understanding facilitates targeted interventions and improvements in educational strategies to enhance self-learning in Python programming among students.

Based on the analysis of factors affecting the self-learning of Python programming among students, strategies need to be designed and implemented to improve Python programming self-learning skills for 10th graders in Da Nang City and other areas. This requires flexible educational strategies and strong support from educators and administrators. Here are some proposed solutions to help 10th-grade students enhance their Python programming capabilities, each with specific steps and implementation methods.

- Proposal 1: Daily Programming Challenges

a) Purpose: Daily programming challenges aim to refine and enhance Python programming skills through continuous practice. This helps students develop programming thinking, quickly and effectively solve problems, and enhance their ability to write creative and accurate code.

b) Content: Daily programming challenges for 10th graders include a variety of programming exercises and challenges, from basic to advanced, using Python. These challenges help students hone their algorithmic thinking, improve coding skills, and effectively apply data structures and algorithms in real-world scenarios.

c) Implementation Steps:

- Step 1: Register on websites like LeetCode, HackerRank, or CodeSignal.

- Step 2: Choose challenges that match skill level and available time.

- Step 3: Program to solve the challenges, then review solutions from others to learn more.

- Proposal 2: Intensive Programming Workshops

a) Purpose: Provide students with essential and in-depth Python programming skills through focused workshops, which can help students quickly adapt and apply knowledge practically. As Resnick et al. (2009) [17] indicated, programming workshops enhance students' creative thinking and problem-solving skills.

b) Content: Implement practical programming workshops with the support of experienced experts and teachers. Workshops will range from basic Python syntax lessons to more complex practices like data handling, web programming, and application development.

c) Implementation Steps:

- Step 1: Identify necessary topics for workshops based on students' learning needs and advancement.

- Step 2: Implement and conduct workshops on weekends or during short breaks so as not to interfere with the regular school schedule.

- Step 3: Evaluate and adjust by collecting feedback from students and teachers after each workshop to assess effectiveness and satisfaction levels.

- Proposal 3: Enhance Use of Digital Educational Resources

a) Purpose: To provide students with diverse learning resources and access Python programming concepts effectively, from basic to advanced levels.
b) Content: Deploy online learning platforms and educational apps to provide students access to high-quality courses and learning materials. These platforms will offer lecture videos, practice exercises, and personal projects.
c)Implementation Steps:
- Step 1: Select and collaborate with reputable online education platforms such as Coursera, Udemy, or Khan Academy.
- Step 2: Develop and integrate Python programming courses into the school curriculum.
- Step 3: Facilitate student participation and completion of exercises and projects provided through the platform.

- Proposal 4: Develop a Peer Learning Support Program

a) Purpose: Encourage students to collaborate and support each other in the learning process, thereby improving programming skills and problem-solving abilities.
b) Content: Create peer learning groups where students can discuss, exchange, and assist each other in programming projects and exercises.
c) Implementation Steps:
- Step 1: Categorize students by programming ability and interest, then form study groups.
- Step 2: Provide necessary learning resources and guidance for each group.
- Step 3: Organize regular meetings for groups to present and discuss their project progress.

- Proposal 5: Develop an 'Hour of Code' Program Integrated into the School Schedule

a) Purpose: Enhance access to and practice of programming for students, thereby improving their self-learning skills and understanding of programming. Research by Guzdial (2015) [18] shows that integrating programming activities into the curriculum can significantly enhance students' understanding and skills.
b) Content: Implement a weekly 'Hour of Code' program where students participate in small programming exercises and projects under teacher guidance. Exercises are designed to suit each student's learning level and needs, from basic to advanced.
c) Implementation Steps:
- Step 1: Develop a curriculum framework that includes programming exercises suitable for each learning level. Select and train teachers to guide effectively during coding hours.
- Step 2: Monitor and evaluate the program's effectiveness through feedback from students and teachers. Adjust content and teaching methods based on evaluation results to ensure the program's appropriateness and effectiveness.

- Proposal 6: Enhance the Use of Online Programming Forums

a) Purpose: Encourage students to use online programming forums such as Stack Overflow and GitHub to learn, solve problems, and interact with the community. Participating in these forums not only helps students improve their programming skills but also develops their information search skills and learning from others, thus establishing a solid foundation for their future programming careers.
b) Content: Introduce and guide students on how to effectively use online platforms like Stack Overflow, GitHub to find solutions, ask questions, and engage in discussions.
c) Implementation Steps:
- Step 1: Organize introductions and training sessions about online platforms, including benefits and basic usage instructions.
- Step 2: Conduct detailed guidance sessions, allowing students to practice creating accounts and asking their first questions.
- Step 3: Provide ongoing support and monitor students' progress on these platforms.

Discussion

In the current educational landscape, understanding the factors affecting 10th graders' ability to self-learn Python programming is crucial. Factors such as perceived usefulness, ease of use, social influence, and facilitating conditions play essential roles in shaping students' attitudes and learning behaviors towards programming. As Venkatesh & Bala (2008) [19] highlighted, perceived usefulness is defined as the subjective probability that using a specific application system will increase one's job performance in an organizational context. When students perceive that learning Python will enhance their job effectiveness or open up future career opportunities, they feel a stronger motivation to learn. This is also supported by the application of Python to solve real-world problems, thus enhancing the perceived applicability of this programming language. Ease of use is also a critical factor, as emphasized by Davis (1989) [20]. When Python is regarded as an easy-to-learn language with clear and concise syntax, students are more likely to engage with it without feeling overwhelmed. This ease of use encourages students to explore more complex aspects of programming as they progress, thereby enhancing their skills and understanding. The proposed solutions, such as mentoring programs, online forums, and the use of open educational resources, aim to enhance the self-learning of Python programming. Mentor support can help students develop problem-solving skills and programming thinking more effectively, as reflected in the study by Garcia & Revano (2021) [21], which indicated that personal guidance could significantly improve students' confidence and learning efficacy. Online forums also facilitate knowledge exchange and timely assistance, supporting them in the self-learning process, which aligns with the findings from Abuhassna et al. (2020) [10] about the positive impact of online communities on learning. Compared to previous studies and models applied in other localities, the solutions proposed in this study focus more on using technology and online communities. Models from studies (Yahya & Jumaat, 2023) [22], have shown the effectiveness of collaborative learning in the programming community, which also supports the necessity of programming clubs and workshops proposed in the current study. The main difference is the emphasis on support from families and the community, which may not have been extensively discussed in previous research.

Conclusion

In this study, we explored factors affecting the self-learning of Python programming among 10th graders in Da Nang City. Analysis of data from 620 students revealed that factors such as individual, family, environment, and society play significant roles in supporting students to develop self-learning skills. Notably, differences in access to educational resources and technology pose a considerable challenge that stakeholders need to address. The development of programming skills, especially Python, is crucial for 10th graders at this time, as technology increasingly becomes an essential part of daily life and the global economy. Self-learning in this field not only helps students enhance personal skills but also boosts their competitiveness in the future labor market.

Based on these findings, we propose several recommendations for stakeholders, including schools, parents, and educational organizations, to cooperate more closely to implement the proposed solutions. Specifically, deploying mentoring programs, developing online learning communities, and providing rich and accessible learning resources are necessary. Additionally, adequate investment in infrastructure and technology at schools is required to ensure that all students, regardless of economic or geographic conditions, have access to high-quality education. We also encourage further research to thoroughly evaluate the effectiveness of intervention programs and explore additional factors that may affect students' self-learning abilities. Through these efforts, we can hope to create an educational environment where each student can fully develop their potential.

References

1. Baker, C. M (2000). Problem-based learning for nursing: integrating lessons from other disciplines with nursing experiences. *Journal of Professional Nursing.* 2000; 16(5): 258-266. PMid:11033935 https://doi.org/10.1053/jpnu.2000.9461
2. Johnson, M., & Smith, K. (2021). Traditional Education in the Digital Age: Perceptions of Educators. *Educational Technology & Society*, 24(3), 152-168.
3. Rini, R., Mujiyati., Sukamto, I., & Hariri, H. (2022). The effect of self-directed learning on students' digital literacy levels in online learning. *International Journal of Instruction*, *15*(3), 329-344. https://doi.org/10.29333/iji.2022.15318a
4. Chusni, M. M., Saputro, S., Rahardjo, S. B., & Suranto. (2021). Student's critical thinking skills through discovery learning model using e-learning on environmental change subject matter. *European Journal of Educational Research,* 10(3), 1123- 1135. https://doi.org/10.12973/eu-jer.10.3.1123
5. Andrea Stocco (2019). *How Artificial Intelligence Can Improve Web Development and Testing: (Invited Paper).* In Companion of the 3rd International Conference on Art, Science, and Engineering of Programming (Programming '19), April 1-4, 2019, Genova, Italy. ACM, New York, NY, USA, 4 pages. (https://doi.org/10.1145/3328433.3328447)
6. Rice, M., & Shannon, L.-J. Y. (2016). Developing project based learning, integrated courses from two different colleges at an institution of higher education: An overview of the processes, challenges, and lessons learned. *Information Systems Education Journal*, *14*(3), 55.
7. Bullock, S. M. (2013). Using digital technologies to support Self-Directed Learning of preservice teacher education. *The Curriculum Journal*, 24, 103-120. https://doi.org/doi.org/10.1080/09585176. 2012.744695
8. Broadbent, J. (2017). Comparing online and blended learner's self-regulated learning strategies and academic performance. *The Internet and Higher Education*, *33*, 24-32. https://doi.org/10.1016/j.iheduc.2017.01.004
9. Jingming, L. (2007) The Expansion of Higher Education and Uneven Access to Opportunities for Participation in It, 1978-2003, *Chinese Education & Society*, 40:1, 36-59. https://doi.org/10.2753/CED1061-1932400103
10. Abuhassna, H., Al-Rahmi, W. M., Yahya, N. et al. (2020). Development of a new model on utilizing online learning platforms to improve students' academic achievements and satisfaction. *International Journal of Educational Technology in Higher Education*, *17*(38). https://doi.org/10.1186/s41239-020-00216-z
11. Hasanah, S. N., Sunarno, W., & Prayitno, B. A. (2020, February). *Profile of Students' Critical Thinking Skills in Junior High Schools in Surakarta*. In 3rd International Conference on Learning Innovation and Quality Education (ICLIQE 2019) (pp. 570-575). Atlantis Press. http://dx.doi.org/10.2991/assehr.k.200129.070.
12. Davis, F. D. (1993). User acceptance of information technology: System characteristics, user perceptions and behavioral impacts. *International Journal of Man-machine Studies*, *38*(3), 475-487. https://doi.org/10.1006/imms.1993.1022
13. Venkatesh, V., Morris, M. G., Davis, G. B, & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *MIS Quarterly*, *27*(3), 425-478. https://doi.org/10.2307/30036540
14. Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2013). Multivariate analysis. Pearson Education Limited.
15. Kaiser, H. F. (1974). An index of factorial simplicity. *Psychometrika*, 39(1), 31-36. https://doi.org/10.1007/BF02291575
16. Fisher, R. A. (1925). *Statistical methods for research workers*. Edinburgh: Oliver and Boyd. http://dx.doi.org/10.1016/B978-044450871-3/50148-0
17. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. (2009). "Scratch: Programming for All." Communications of the ACM, 52(11), 60-67. https://doi.org/10.1145/1592761.1592779
18. Mark Guzdial. (2015) Learner-Centerd Design of Computing Education: Research on Computing for Everyone 8(6):1-165. https://doi.org/10.1007/978-3-031-02216-6
19. Venkatesh, V. & Bala, H. (2008). "Technology Acceptance Model 3 and a Research Agenda on Interventions." Decision Sciences, 39(2), 273-315. https://doi.org/10.1111/j.1540-5915.2008.00192.x
20. Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology.MIS Quarterly, 13(3), 319-340. https://doi.org/10.2307/249008
21. Garcia, M. B., Revano, T. F. (2021). *Assessing the Role of Python Programming Gamified Course on Students' Knowledge, Skills Performance, Attitude, and Self-Efficacy.* 2021 IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM).

22. Yahya, A. H., & Jumaat, N. F. (2023). The Effectiveness of Online Collaborative Learning towards Programming Skills among Undergraduate Students in Learning Programming. *Malaysian Journal of Social Sciences and Humanities (MJSSH)*, *8*(7), e002398. https://doi.org/10.47405/mjssh.v8i7.2398
23. Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika, 16*(3), 297-334.
24. Fraser, B. J. (2014*). Classroom learning environments: Historical and contemporary perspectives.* In Handbook of research on student engagement (pp. 315-328). Springer, Boston, MA.
25. Huynh Kim Cuong (2023). Solutions to improve students' ability to work in pairs and achieve higher results. *Journal for Educators, Teachers and Trainers*, *14*(3), 132-137. https://doi.org/10.47750/jett.2023.14.03.016
26. Iqbal, S., Mathew, R., Tawafak, R.M., Khan, I. (2019*). Gender Difference in Perceiving Algorithmic Thinking In An Introductory Programming Course*, EDULEARN19 Proceedings, pp. 8246-8254. https://doi.org/10.21125/edulearn.2019.2042)
27. Jones, D. E., & Kessler, R. (2018). Technology and educational equality: Bridging the digital divide in education. *Journal of Educational Research*, *112*(1), 94-100.
28. Ryan, A. M., & Patrick, H. (2001). The classroom social environment and changes in adolescents' motivation and engagement during middle school. *American Educational Research Journal*, *38*(2), 437-460. https://doi.org/10.3102/00028312038002437
29. Singh, P. K., & Nguyen, T. V. (2019). Challenges in educational research in emerging economies and the global south. *International Journal of Educational Development*, *68*, 130-135.
30. Thompson, R. L., Higgins, C. A., & Howell, J. M. (1991). Personal computing: Toward a conceptual model of utilization. MIS Quarterly, 15(1), 125-143. https://doi.org/10.2307/249443.
31. Wilder, S. (2014). Effects of parental involvement on academic achievement: A meta-synthesis. *Educational Review, 66*(3), 377-397. https://doi.org/10.1080/00131911.2013.780009.
32. Yeh, H. W. (2022). Effects of Diagnostic Tests on Self-Directed Language Learning Development. *International Journal of Computer-Assisted Language Learning and Teaching (IJCALLT),* *12*(1), 1-15. http://doi.org/10.4018/IJCALLT.291105.
33. Zimmerman, B. J. (2000). Self-efficacy: An essential motive to learn. *Contemporary Educational Psychology*, *25*(1), 82-91. https://doi.org/10.1006/ceps.1999.1016.