Unravelling Physical Manipulatives in the Learning of Coding: Exploring Student Perceptions in the Nexus of Computer Programming and Robotics

Reginald Gerald Govender* and Desmond Wesley Govender

School of Education, Mathematics and Computer Science Education Cluster, University of KwaZulu-Natal, South Africa
Email: govenderR4@ukzn.ac.za (R.G.G.); govenderD50@ukzn.ac.za (D.W.G.)
*Corresponding author

Manuscript received September 5, 2024; revised September 23, 2024; accepted November 7, 2024; published March 20, 2025

***Abstract*—In the digital age, computer programming is a valuable skill. However, novice programmers typically encounter problems and challenges that lead to negative reactions and dropout. This study examines how a group of university computer-registered students without programming experience learn computer programming through the coding of robots. Instead of traditional block-based programming to teach computer programming, the study used a robotic element and text-based programming to develop prototypes offering live autonomous code output. A research model was created and tested to determine contributing factors to how students perceived computer programming through coding robots. Belief, Interest, Mathematics, Knowledge, Confidence and Motivation formed the latent variables. The research hypotheses are: 1) Belief affects Confidence, 2) Confidence affects Knowledge, 3) Interest affects Confidence, 4) Mathematics affects Knowledge, and 5) Motivation affects Confidence. Participants included seventy-five students who completed a 5-point, 30-item Likert scale survey to assess their robotics computer programming experience at a University in South Africa. Partial least square-structural equation modelling was conducted to investigate the relationship between latent variables. The model explains the relationship between Belief and Confidence, Confidence and Knowledge, Interest and Confidence, Mathematics and Knowledge, and Motivation and Confidence. Results demonstrate that students studying text-based programming directly with the robotic element were successful. As participants saw their code run on the prototype, robot coding made text-based coding easier to understand. Coding structures were clarified by using robotics to make computing programming concrete. The learning of computer programming integrated through the building of prototypes resulting in autonomous robots enhances the learning experience of text-based code. This research contributes empirical evidence and elucidates the factors that may influence learning satisfaction of text-based computer programming through coding robotics.**

***Keywords*—coding and robotics, computer programming, physical manipulatives**

I. Introduction

Computer programming can be challenging since students face a steep learning curve [1, 2], likely without prior knowledge [3]. However, students interested in computer programming are likely to be engaged and motivated to learn intrinsically. Research has shown that students with a strong interest in programming will likely invest time and effort into learning the required concepts and skills [4–6]. At the same time, student interest plays a crucial role in learning computer programming. Individuals genuinely interested in computer programming are more likely to be motivated to learn and explore the subject further. Students' interest in programming can be nurtured by providing opportunities to explore real-world applications through educational robotics [2, 3]. Such physical manipulatives allow students to apply their programming knowledge in real-world scenarios, which can significantly enhance their engagement and enthusiasm for the subject [7]. Therefore, it is crucial to understand different approaches that can be used when teaching programming. There is no approach to introducing programming that guarantees the acquisition of programming expertise. As a result, the following questions arise: What are the best practices for learning computer programming? If students consider programming challenging, the researcher wonders: How might learning computer programming be promoted? Hence, this study sets out to test a proposed model by examining the predictors contributing to one's knowledge of programming through robotics, thereby seeking to answer the following: What are students' perceptions of robotics when learning to program?

II. Literature Review

Motivation plays a crucial role in learning, as when students are motivated to learn programming, they are more likely to set goals, persevere through challenges, and engage in deeper learning [6, 8]. Studies have shown that using robots as a tool in programming education can significantly increase student motivation [3, 7, 9, 10]. Moreover, it has been observed that students were happier, and their motivation was higher in programming courses using robots than in courses using classical programming education methods [11]. Seminal research by South African mathematician and computer scientist Seymour Papert postulates that educational robotics promotes constructionism in that students construct knowledge rather than passively assimilate information [12, 13].

Confidence plays a crucial role in the learning of computer programming. When students have confidence in their abilities, they are more likely to take risks, try new things, and persist when facing challenges. Students with higher confidence levels tend to perform better in programming tasks and are more likely to use problem-solving strategies [14]. Confidence is a significant factor that affects the learning of computer programming. Belief in one's problem-solving ability is crucial to learning computer programming. When students believe in their problem-solving abilities, they approach programming tasks with confidence and a positive mindset. This belief can lead to increased persistence and resilience when facing challenges, ultimately enhancing learning outcomes [15]. Belief in problem-solving abilities can be fostered by providing students with opportunities to practice and succeed in solving programming problems. By gradually increasing the difficulty level of problems and providing appropriate support and feedback, students can develop a sense of efficacy and confidence in their problem-solving skills [16].

Mathematics is an essential skill due to its practical applications in physics, engineering, economics, computer science, and more [17, 18]. Mathematics forms the foundation of many programming concepts, such as algorithms, logic, and problem-solving techniques. Additionally, math skills are essential for handling complex calculations and data manipulation in programming tasks. However, research has found conflicting views about the relationship between math and programming ability. Some studies suggest that math can enhance problem-solving skills and logical thinking, which are crucial in programming [19, 20], and it impacts computer programming [21–23]. On the other hand, some argue that it is not a prerequisite for becoming a successful programmer or problem solver and may have a negative impact [24–26].

III. Materials and Methods

A. Selection of Constructs/Latent Variables

Regression analysis carried out in a study by Tsai [27] revealed that *Mathematics* was a positively correlated predictor of programming knowledge. The correlation was found in a pre-test and post-test. Additionally, an individual's *motivation* to learn Computer Programming is a crucial factor in their acquisition of programming knowledge. As Carbone [28] elaborates, those students who exhibited motivation "usually undertook to learn programming in their own time, sometimes prior to the course commencing, working hard at developing their skills". Computer Programming can be perceived as complex and intimidating, especially text-based language-specific programming. A study carried out by Blanchard [29] found that "hybrid programming[1] environments can help to transition students from blocks to text-based programming while minimising negative perceptions of programming". Thus, a student's *belief* about coding is important to their programming knowledge. Workshops, being practical and hands-on, have the potential to build and maintain students' *interest* in Computer Programming. As remarked by Biggers [30], "highly interactive, hands-on introductory CS courses…provide a broader overview of potential CS". An individual's *interest* in coding can thus affect their knowledge of programming. As far back as 2001, Byrne and Lyons showed that students enrolled in a programming course are prone to show anxiety, such as being less confident—especially if they do not have prior computer experience [31]. Similarly, from a comparative study between coding IDEs, Daly [32] commented that "learning abstract programming concepts and programming in an environment … can cause students to become frustrated, lose confidence". Therefore, *confidence* in learning coding can potentially affect the development of programming knowledge.

Based on the corpus of the literature reviewed, the survey design took into account the following six constructs in alignment with the proposed model that have been shown to influence knowledge of coding: student *confidence* in their ability to learn programming, student *interest* in programming, student *motivation* to use Robotics, student intrinsic *belief* that they can solve problems, student perception of mathematical influence on programming and student *knowledge* of programming through the use of Robotics.

B. Participants

Simple random sampling was employed on a homogenous population. The population's characteristics were that they were subjects at a University in KwaZulu-Natal, South Africa, with no prior computer programming experience. An invitation detailing the study, anonymity, and the project's voluntariness was issued. Based on the response to the invite, the sample comprised 75 students.

C. Procedures

A series of workshops took place during the semester using educational robotic kits to introduce the basics of computer programming. The kits used were the Arduino and Lego Mindstorm series, focusing on text-based coding in Python. The workshops covered the following topics: introduction of the kits, sensors, data structures, data types, variables, string and character handling, math handling, iterations structures, selection structures and nested structures. Prototype design was the ultimate objective, allowing students to build-code-test by implementing and integrating real-world scenarios. Students were asked to complete a survey at the end of the semester.

D. Instrument

A 5-point, 30-item Likert scale survey was administered at the semester's end and captured the immediate thoughts of their Computer Programming experience using the robotic kit. Items were formed based on the six constructs to collate and structure the survey. Each construct comprised five items. Hence, the survey consisted of 30 items. The internal consistency of the items was examined by the composite reliability (discussed under indicator reliability), resulting in 19 items being retained. Composite Reliability is an alternative to Cronbach's alpha. Hair [33] points out that a drawback of Cronbach's alpha is that it implies all indicator loadings in the population are equal, which is also known as tau-equivalence.

E. Hypothesis and Model Development

In PLS-SEM, constructs are known as Latent Variables (LV), and items are known as indicators. The model created (Table 1) follows the principles of a reflective model since, as Garson [34] explains, in such models, "indicators are a representative set of items which all reflect the latent variable they are measuring". This is observed in the model created during the initial stages of the study (Fig. 1), where all six LVs offer loadings onto the respective five indicators. Reflective models allow the omitting or dropping of indicators that do not matter while sustaining the meaning of the LV [34]. It is important to note that omitting indicators

[1]A mix of text-base and block-base environments

that are not significant would be essential in producing a model that makes meaning and sense.

Table 1. Measurement scale

Latent Variables	Items code	Items
Belief	Belief1	I could come up with a suitable strategy for a given programming project in a short time.
	Belief2	I could manage my time efficiently if I had a pressing deadline on a programming project.
	Belief3	I could find ways of motivating myself to program, even if the problem area was of no interest to me.
	Belief4	I could complete a programming project if someone showed me how to solve the problem first.
	Belief5	I could complete a programming project if I could call someone for help if I got stuck.
Interest	Interest1	I hope to use programming sometime in my future.
	Interest2	The challenge of solving problems using programming does not appeal to me.
	Interest3	I can use the thinking developed when programming in my daily life.
	Interest4	I think computer science (programming) is interesting.
	Interest5	I would voluntarily take additional Computer Science courses to learn programming.
Motivation	Motivation1	I was more interested in the robotic element (microcontroller) than the programming.
	Motivation2	I found programming attractive through the use of the microcontroller.
	Motivation3	The use of the microcontroller held my attention for longer.
	Motivation4	Easier for me to comprehend and retain information in coding through the use of the physical device (microcontroller).
	Motivation5	Programming the microcontroller was a positive experience for me.
Mathematics	Mathematics1	I think mathematics is not needed to be able to program.
	Mathematics2	I can translate the mathematical content into a programming code.
	Mathematics3	I think basic mathematics is needed to be good at programming.
	Mathematics4	I think you would need more than basic mathematics knowledge to program.
	Mathematics5	I think the mathematics involved in programming is technical.
Confidence	Confidence1	I am comfortable with learning programming concepts.
	Confidence2	I have little self-confidence when it comes to programming.
	Confidence3	I can learn to understand programming concepts on my own.
	Confidence4	I think I can achieve good grades in programming.
	Confidence5	I am confident that I can solve problems by using programming.
Knowledge	Knowledge1	The microcontroller allowed for the understanding of programming.
	Knowledge2	I would have understood the programming concepts without the robotic element (microcontroller).
	Knowledge3	The microcontroller provided a visual aid of what my programming was doing.
	Knowledge4	I could rewrite lengthy and confusing portions of code to be readable and clear.
	Knowledge5	I could write a program that someone else could comprehend and add features to it later on.

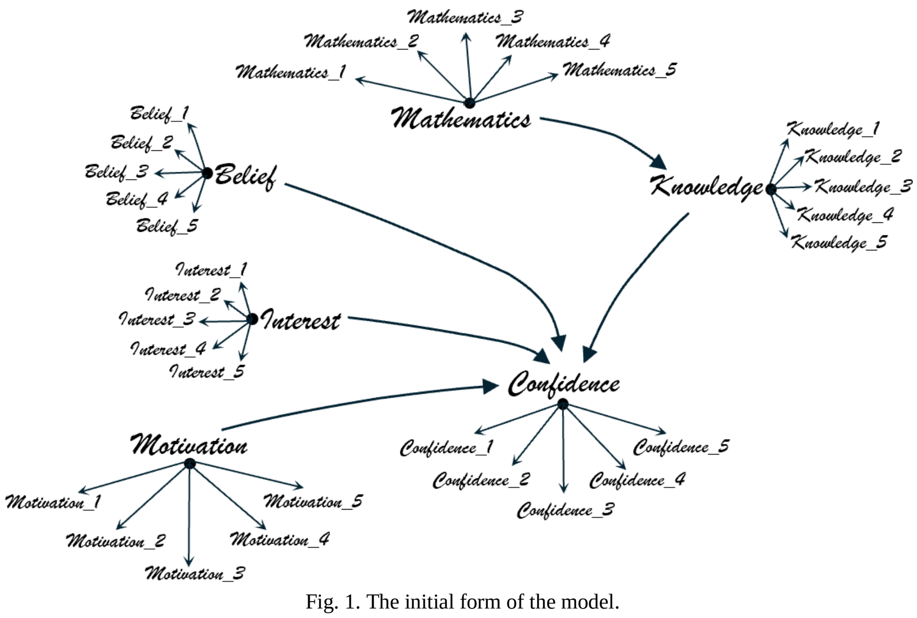

Fig. 1. The initial form of the model.

Fig. 1 depicts the model in the early stages before running any analysis, showing all LVs and indicators. *Knowledge* is referred to as an endogenous LV. An endogenous latent variable is an LV informed by at least one other LV [34, 35]. In other words, graphically, the model has at least one incoming arrow from another LV. The LVs: *belief, interest, motivation*, and *Mathematics* are referred to as exogenous variables since any other LV does not inform them. Hence, they do not have any incoming arrow/s from any other LV/s.

The influence of *Mathematics* on coding holds mixed views regarding its effect on coding *knowledge* [21–25]. An individual's *confidence* is informed by their *belief*, *interest*, and *motivation* [36–39]. While the latter factors may not directly affect *knowledge*, they have the potential to affect it indirectly through *confidence* [40, 41]. Hence, *confidence* is referred to as a mediating variable, an intervening variable.

Table 2. Set of hypotheses based on model

Hypothesis	Relationship
H1	Belief → Confidence
H2	Confidence → Knowledge
H3	Interest → Confidence
H4	Mathematics → Knowledge
H5	Motivation → Confidence

Note: The alternative hypothesis (Ha) for each relationship found in Table 2 is that there is no relationship between the LVs.

Through robotics, *knowledge* of coding is a dependent LV informed by the following possible contributing factors: *belief*, *interest*, *motivation*, *confidence*, and *Mathematics*. The study will determine if the following relationships exist through the SEM-PLS model; hence, they hold true (Table 2).

F. Sample Size

The minimum sample size estimation method in PLS-SEM follows the 10 times rule [35, 42]. The rule applied to the reflective conceptual model in this study would be:

Sample size > 10 × the largest no. of structural paths directed at a particular LV in the structural model. The conceptual model (Fig. 1) indicates that the largest number of structural paths directed to a particular LV is three (i.e., *belief*, *interest* and *motivation* informing *confidence*). Therefore, the sample size aligns with the ten times rule: 75 (sample size) > 10 ×3 (largest no. of structural paths) = 75 > 30.

IV. Result and Discussion

This study applied Partial Least Squares-Structural Equation Modelling (PLS-SEM) to assess the data generated from the survey. PLS-SEM is commonly used in business industrial research and analysis; however, the statistical modelling technique has gained popularity in education research [43, 44]. Ramli [45] found that PLS-SEM analysis offers fewer contradictory results than regression analysis despite PLS-regression models being a subset of PLS-SEM models. Garson [34] explains that PLS-SEM models differ from regression models as they are "path models in which some variables may be effects of others while still be causes for variables later in the hypothesized causal sequence". PLS-SEM analysis consists of two parts: firstly, it examines the measurement model, which consists of indicator reliability, convergent reliability, and discriminant validity. Secondly, the structural model assesses collinearity issues, path coefficients, the significance of the relationships, level of R^2, effect size (f^2) and predictive relevance (Q^2).

A. Measurement Model

The measurement model is presented in three parts: indicator reliability, convergent reliability and discriminant validity. Indicator reliability examines the measure and validity of the reflective indicator loadings, Cronbach's Alpha (CA) and rho_A (ρA). The convergent reliability examines the Average Variance Extracted (AVE) and internal consistency. The evaluation of discriminant validity considers the cross-loadings, Fornell and Larcker criterion and Heterotrait-Monotrait ratio of correlations (HTMT).

1) Indicator reliability

The PLS algorithm was executed on the model (Fig. 1) with an initial analysis of 300 iterations and later with a maximum of 500 iterations, resulting in the outer indicator loadings (Fig. 2). The loadings can be considered a form of item reliability in reflective models; as Garson [34] describes, "the closer the loadings are to 1.0, the more reliable that latent variable". He goes on to express that path loadings for such a model should be > 0.7 [34], while Hulland [46] recommends that reflective indicator loadings > 0.5 show that the indicator is a good measurement of the LV. A more refined and applied criterion to this study is by Hair *et al.* [47], who propose that an indicator loading in the range of 0.40 to 0.70 may be dropped only if it improves Composite Reliability (CR). Therefore, outer loadings of 0.7 or higher are considered highly approved, while 0.5 is deemed acceptable [48]. Fig. 2 depicts the outer loadings of indicators that meet the criteria of > 0.7 or are retained because they do not improve CR when discarded.

The CA value evaluates the reliability of the set of indicator items. Therefore, it measures the extent to which all the LVs in the model are positively related to each other. An alpha value greater than 0.7 ($\alpha > 0.7$) is acceptable [49]. Using rho_A (ρA) is a more consistent measure of reliability than Cronbach's Alpha. As Dijkstra and Henseler [50] describe, ρA is measured the same as Cronbach's Alpha and has a better reliability measure than Cronbach's Alpha in SEM since ρA is based on the loadings rather than the correlations observed between the variables.

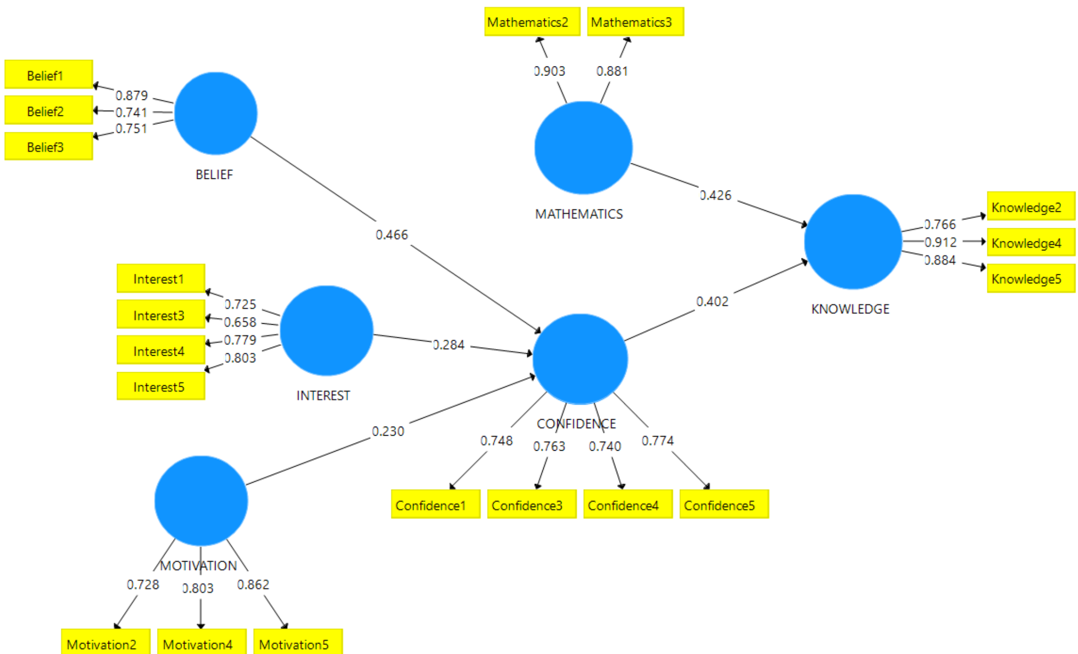

Fig. 2. Model showing valid path loadings.
Note. The weighting scheme was initially set to a maximum of 300 iterations and later set to a maximum of 500 iterations.

2) *Convergent reliability*

One of the measures of convergent reliability is Average Variance Extracted (AVE). AVE is comparable to the proportion of variance explained in factor analysis. AVE values are between 0 and 1; an AVE > 0.5 is desired [51, 52]. Internal Consistency is assessed by validating the Composite Reliability (CR), which measures the reliability of the indicators where values are between 0 and 1. CR values greater than 0.7 (CR > 0.7) prove to show adequate consistency [53]. Chin [54] proposed that internal consistency is measured through the CR, known as Dillon-Goldstein's rho or Jöreskog's rho. CR is a favoured alternative to CA as a test of convergent reliability in a reflective model [34]. Demo *et al.* [55] propound the view that CR offers a more reliable measure than CA in SEM. The findings support Dijkstra and Henseler [50] when examining (Table 3), the cut-off value of ρA (Rho A) > 0.7 ensures Composite Reliability (CR > 0.7).

3) *Discriminant validity*

Discriminant validity is the extent to which an LV is truly distant from other LVs, which implies that the LV is unique. Discriminant validity, also known as vertical collinearity, validates the subjective independence of every indicator on the LV. The cross-loadings criterion helps reduce the presence of multicollinearity amongst the LVs by denoting that the AVE of an LV should be higher than the square correlations between the LV and all other variables [52, 54, 56]. In other words, the loadings of an indicator on its assigned LV should be a higher value than its loadings on all other associated LV values (refer to Table 4, values in bold are greater than horizontal associated values).

The Fornell and Larcker criterion is that the AVE of an LV should be higher than the squared correlations between the LV and all other variables [52, 54, 56]. The software processes this calculation in which the diagonals are the square root of the AVE of the LVs (Table 5) and should be the highest in any associated column or row.

Table 3. Measurement model including LV (construct) reliability and validity

Items	Loadings	AVE	CR	CA	Rho A
Belief1	0.879		0.834	0.703	0.731
Belief2	0.741	0.628			
Belief3	0.751				
Interest1	0.725	0.553	0.831	0.729	0.725
Interest3	0.659ǂ				
Interest4	0.779				
Interest5	0.803				
Motivation2	0.728	0.639	0.841	0.716	0.730
Motivation4	0.803				
Motivation5	0.862				
Mathematics2	0.903	0.795	0.886	0.744	0.751
Mathematics3	0.881				
Confidence1	0.748	0.572	0.842	0.751	0.750
Confidence3	0.763				
Confidence4	0.740				
Confidence5	0.774				
Knowledge2	0.766	0.733	0.891	0.821	0.867
Knowledge4	0.912				
Knowledge5	0.884				

Note: a) Indicator items removed: Belief 4, Belief 5, Interest 2, Motivation 1, Motivation 3, Mathematics 1, Mathematics 4, Mathematics 5, Confidence 2, Knowledge 1, Knowledge 3. b) All item loadings > 0.5 indicate indicator reliability [46, 47]. c) All Average Variance Extracted (AVE) > 0.5 indicates convergent reliability [51]. d) All Composite. e) Reliability (CR) > 0.7 indicates internal consistency [53]. f) All Cronbach's Alpha (CA) > 0.7 [49]. g) All rho_A (ρA) > 0.7 [50]. h) ǂ Retained because it did not change the CR when dropped/removed.

Table 4. Discriminant validity-indicator item cross-loading

Items	Belief	Confidence	Interest	Knowledge	Mathematics	Motivation
Belief1	**0.881**	0.707	0.505	0.589	0.879	0.624
Belief2	**0.741**	0.574	0.371	0.566	0.513	0.493
Belief3	**0.751**	0.488	0.370	0.344	0.546	0.249
Confidence1	0.524	**0.748**	0.684	0.516	0.444	0.603
Confidence3	0.604	**0.763**	0.412	0.557	0.580	0.402
Confidence4	0.664	**0.740**	0.416	0.502	0.574	0.548
Confidence5	0.492	**0.774**	0.581	0.502	0.426	0.590
Interest1	0.342	0.463	**0.725**	0.466	0.332	0.579
Interest3	0.599	0.594	**0.658**	0.473	0.570	0.478
Interest4	0.211	0.470	**0.779**	0.296	0.131	0.440
Interest5	0.361	0.499	**0.803**	0.408	0.250	0.590
Knowledge2	0.288	0.469	0.472	**0.766**	0.346	0.591
Knowledge4	0.652	0.588	0.489	**0.912**	0.653	0.528
Knowledge5	0.630	0.674	0.492	**0.884**	0.705	0.546
Mathematics2	0.628	0.496	0.314	0.647	**0.903**	0.330
Mathematics3	0.879	0.707	0.505	0.589	**0.881**	0.624
Motivation2	0.414	0.547	0.488	0.550	0.530	**0.728**
Motivation4	0.426	0.492	0.526	0.458	0.255	**0.803**
Motivation5	0.576	0.646	0.660	0.512	0.459	**0.862**

Note: All item loadings on its assigned latent variable (bold values) are higher than those on all other latent variables. In other words, all the indicator's outer loading on the associated LV is greater than all its loadings on other LVs; therefore, the cross-loadings criterion is fulfilled [52, 54, 56].

Table 5. Discriminant Validity- Fornell and Larcker criterion

Items	Belief	Confidence	Interest	Knowledge	Mathematics	Motivation
Belief	**0.793**					
Confidence	0.755	**0.756**				
Interest	0.531	0.694	**0.743**			
Knowledge	0.643	0.686	0.562	**0.856**		
Mathematics	0.838	0.668	0.454	0.694	**0.892**	
Motivation	0.598	0.710	0.705	0.635	0.527	**0.800**

Note: Fornell and Larcker's criterion has been met in all instances since all values (in bold) are greater than their associated values (the values in bold forming the diagonal are the highest compared to associated values across and below).

4) Heterotrait-Monotrait ratio of correlations, bootstrapping and normality

The Heterotrait-Monotrait ratio of correlations (HTMT) was developed to address the insensitivity of the Fornell and Larcker criterion and cross-loading. Determining if discriminant validity is achieved using HTMT requires bootstrapping. Although PLS-SEM assumes that the data is not normal, unlike CBS-SEM (Covariance Based Structural-Equation Modelling), this study will still confirm that the data is not normal before proceeding to bootstrap. Bootstrapping is a nonparametric procedure that allows testing statistical significance through a method that uses random sampling, mimicking the sampling process- resampling method. It can be applied to regression models, giving insight into how variable the model parameters are [57]. Bootstrapping estimates the spread, shape, and bias of the population's sampling distribution. The observed sample is treated as a representation of the population. Bootstrapping creates a large, pre-specified number of samples. Every time sampling happens during bootstrap, the same number of cases as the original sample will be analysed; thus, N bootstrap → n samples [54]. The cut-off value for univariate skewness is ±1, and kurtosis is ±7 [58, 59]. The cut-off value for Mardia's multivariate skewness is ±1, and kurtosis is ±20 [60].

The findings from the normality tests (Table 6) show that the kurtosis values are within bounds; however, the skewness for LVs: *Belief*, *Interest* and *Motivation* are out of bounds. Further examining Mardia's coefficient (Table 7), the values of skewness and kurtosis are out of the respective bounds (Skewness $b > 1$; Kurtosis $b > 20$). The normality assessment deduces that the data does not follow the normal distribution; hence, bootstrapping can be applied [61].

As a result of bootstrapping, the discriminant validity based on the Heterotrait-Monotrait ratio of correlations (HTMT) can be examined. HTMT estimates the correlation between the LV based on the average Heterotrait-Heteromethod correlation [62]. HTMT is assessed by examining the Confidence Interval- Upper Limit (CI-UL) and is expected to be less than 0.90 (at the 95% Confidence Interval). Therefore, a CI-UL value higher than 0.9 indicates a lack of discriminant validity. Ringle [63] purports that discriminant validity is not established if the CI-UL value is above 1. As a statistical test- testing of the null hypothesis (H_0: HTMT<1) versus the alternative (H_a: HTMT ≥1) [62], $HTMT_{95\%\ Confidence\ Interval}$ contains the value one or above→; hence, no discriminant validity.

Table 6. Assessing Normality-Univariate and multivariate skewness and kurtosis

Items	Skewness	SE_skew	Kurtosis	SE_kurt
Belief	1.0581	0.2774	1.5678	0.5482
Confidence	0.6666	0.2774	0.5753	0.5482
Interest	1.4769	0.2774	2.7626	0.5482
Knowledge	0.3320	0.2774	−0.1976	0.5482
Mathematics	0.7718	0.2774	1.0837	0.5482
Motivation	1.3041	0.2774	2.1185	0.5482

Note: a) Sample size: 75; b) Number of variables: 6; c) SE_skew= Standard error skewness; d) SE_kurt= Standard error kurtosis

Table 7. Assessing Normality-Mardia's multivariate skewness and kurtosis

Coefficient	b	z	*p*-value
Skewness	11.96683	149.58536	1.84E-10
Kurtosis	52.12194	1.821659	6.85E-02

Note: b = Mardia's coefficient for skewness and kurtosis.

An initial assessment was conducted using a smaller number of complete bootstrapping with subsample sizes of 500, 1000 and 3000 with parallel processing. A large subsample size of 5000 was used during bootstrapping for final result preparation. The findings (Table 8) indicate the null hypothesis has failed to be rejected; hence, discriminant validity has been established, CI-UL < 0.9.

Table 8. Discriminant validity-HTMT

Relationship	Original Sample (O)	Sample Mean (M)	CI LL 5.00%	CI UL 95.00%
Belief → Confidence	0.466	0.470	0.353	0.592
Confidence → Knowledge	0.402	0.397	0.216	0.576
Interest → Confidence	0.284	0.292	0.129	0.454
Mathematics → Knowledge	0.426	0.432	0.261	0.600
Motivation → Confidence	0.230	0.216	0.067	0.362

Note: a) Complete bootstrapping performed. b) Set at 5000 subsamples. c) Parallel processing. d) H_0 holds since all CI-UL < 0.9 [62].

Table 9. Structural model-variance inflated factor

Items	Belief	Confidence	Interest	Knowledge	Mathematics	Motivation
Belief		1.618				
Confidence				1.807		
Interest		2.066				
Knowledge						
Mathematics				1.807		
Motivation		2.310				

Note: All VIF values are within the prescribed tolerance ranges (VIF ≥ 5, [64]; VIF ≥ 3, [65]).

B. Structural Model

The structural model examines horizontal collinearity. Therefore, assessing the structural model results enables determining the model's capability to predict one or more target LV/s (construct/s). The study presents the structural model in six parts: collinearity issues, path coefficients, the significance of the relationships, level of R^2, effect size (f^2) and predictive relevance (Q^2).

1) Collinearity issues

Collinearity arises when two indicators are highly correlated. Collinearity among LVs is assessed through the Variance Inflated Factor (VIF). A VIF value of greater than or equal to five (VIF ≥ 5) indicates a potential collinearity problem [64]. Meanwhile, a more stringent guideline purported by Diamantopoulos and Siguaw [65] is that a VIF ≥ 3 indicates potential collinearity problems. Table 9 shows the evaluation of the VIF values, where each set of predictor LV is assessed separately for each subpart of the structural model. All VIFs are within the prescribed guidelines [64, 65], meaning there is no strong indication of collinearity issues.

2) Path coefficients

Path coefficients are the coefficients linking LVs in the structural model. The coefficient represents the hypothesised relationship of the relationship strength, hence the significance of the relevance of the relationships. Accordingly, the primary way to compare the strength of relationships is to examine the path coefficients. The path coefficients indicate to what extent an independent variable affects a dependent variable (through bootstrapping, the significance of the relationships is examined- *observed in the next subheading*). Path coefficients vary between −1 and +1; coefficient values closer to +1 indicate a robust positive relationship (and vice versa for negative values). Higher values denote more robust (predictive) relationships between the LVs. When the value is closer to 0, it signifies a weak relationship and is not statistically significant.

There are three types of effects. Firstly, a *direct effect* is a relationship linking two LVs with a single arrow between the two. Secondly, an *indirect effect* is a sequence of relationships with at least one intervening LV involved. Third, the *total effect* is the sum of the direct and indirect effects linking two LVs. It is important to note that the conceptual model created and tested in this study (Fig. 1) is designed based on a direct effect relationship.

The path coefficient values (Table 10) do not exhibit high values but indicate all positive relationships between IVs and DVs. Fig. 3 visually depicts this, showing that all positive relationships exist.

3) Significance of the relationships-t values and p values

The significance of the relationships is further assessed through bootstrapping (i.e., examining whether the effect of a specific IV on a certain DV is significant). The bootstrapping analysis evaluates the direct effects of all the hypothesised relationships represented by statistical testing of the hypothesis (Table 2). Determining whether a coefficient is significant depends on its Standard Error (SE) obtained by bootstrapping that computes the empirical *t*-values and *p-values* for all structural paths. When an empirical *t*-value is larger than the critical value, it can be concluded that the coefficient is statistically significant at a certain error probability. Commonly used critical values for a two-tailed test are 1.65 at a significant level of 10%, 1.96 at a significant level of 5% and 2.58 at a significant level of 1%. The confidence level is equivalent to 1; in humanities, it is typical to adopt a significance level of 0.05, which corresponds to a confidence level of 95%.

Table 10. Path coefficients

Items	Belief	Confidence	Interest	Knowledge	Mathematics	Motivation
Belief		0.466				
Confidence				0.402		
Interest		0.284				
Knowledge						
Mathematics				0.426		
Motivation		0.230				

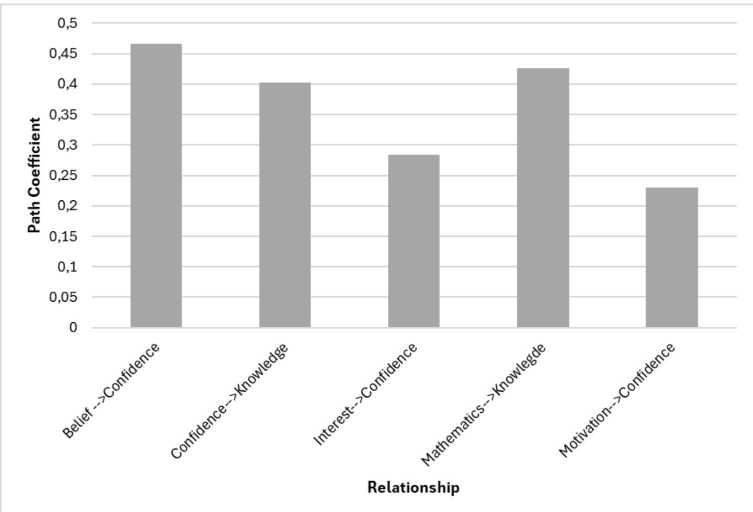

Fig. 3. Visual representation of path coefficients.

If $t_{0.05} > 1.96$ (for a 2-tailed test, critical value = 1.96), the hypothesis is supported [66]. Hair *et al.* [35] suggest assessing beta values (β) and the corresponding t values through a bootstrapping procedure with a resample of 5000. The individual path coefficient is interpreted as the standardised coefficient in an ordinary least squares (OLS) regression. A one-unit change of an exogenous construct changes the endogenous construct by the size of the path coefficient when everything else remains constant [64].

Table 11. Direct relationships for Hypothesis testing

Hypothesis	Relationship	Std Beta	Std Error	*t* value	*p*-value	Decision	95%CI LL	95%CI UL
H1	Belief → Confidence	0.470	0.073	6.352**	0.000	✓	0.353	0.592
H2	Confidence → Knowledge	0.397	0.109	3.689**	0.000	✓	0.216	0.576
H3	Interest → Confidence	0.292	0.100	2.836**	0.005	✓	0.129	0.454
H4	Mathematics → Knowledge	0.432	0.104	4.103**	0.000	✓	0.261	0.600
H5	Motivation → Confidence	0.216	0.089	2.578*	0.010	✓	0.067	0.362

Note: a) ** $p < 0.01$, * $p < 0.05$; b) ✓ means decision is supported; c) All relationships supported.

The p-values are used to assess the significance levels. When assuming a significance level of 5%, the p-value must be smaller than 0.05 to conclude that the relationship under consideration is significant. It is noted that most p-values meet the condition when assuming a significance level of 1%, where the p-value is smaller than 0.01, which concludes that the relationship under consideration is significant not only at a 5% level but also at a 1% level (Table 11). Similar research found that tasks involving robots have improved students' engagement, interest, attitude, and motivation [67–69].

4) Level of R^2

R square (R^2) is the coefficient of determination, which measures the proportion of variance in a latent endogenous variable explained by the other exogenous expressed as a percentage [54]. Hence, R^2 measures the model's predictive accuracy, representing the amount of variance in the endogenous constructs explained by all exogenous constructs linked to it. R^2 values range between 0 and 1, with higher values indicating higher levels of predictive accuracy.

It is considered that values of $R^2 \approx 0.25$: weak, $R^2 \approx 0.50$: moderate, and $R^2 \approx 0.75$: substantial [64, 70]. Chin [54] articulates R^2 values of 0.67, 0.33 and 0.19 as substantial, moderate and weak. The adjusted R2 values are interpreted similarly to the R^2 square values, as the adjusted R^2 controls for model complexity when comparing different model set-ups [71, 72].

Table 12. R Square (R^2) values

	R Square	R Square Adjusted	Outcome
Confidence	0.713	0.701	Substantial- Moderate
Knowledge	0.571	0.559	Moderate- Substantial

Note: R^2 and R^2 adjusted values for Confidence are substantial, and Knowledge is moderate- $R^2 \approx 0.19$- 0.25: weak, $R^2 \approx 0.33$- 0.50: moderate and $R^2 \approx 0.67$- 0.75: substantial [54, 64, 70].

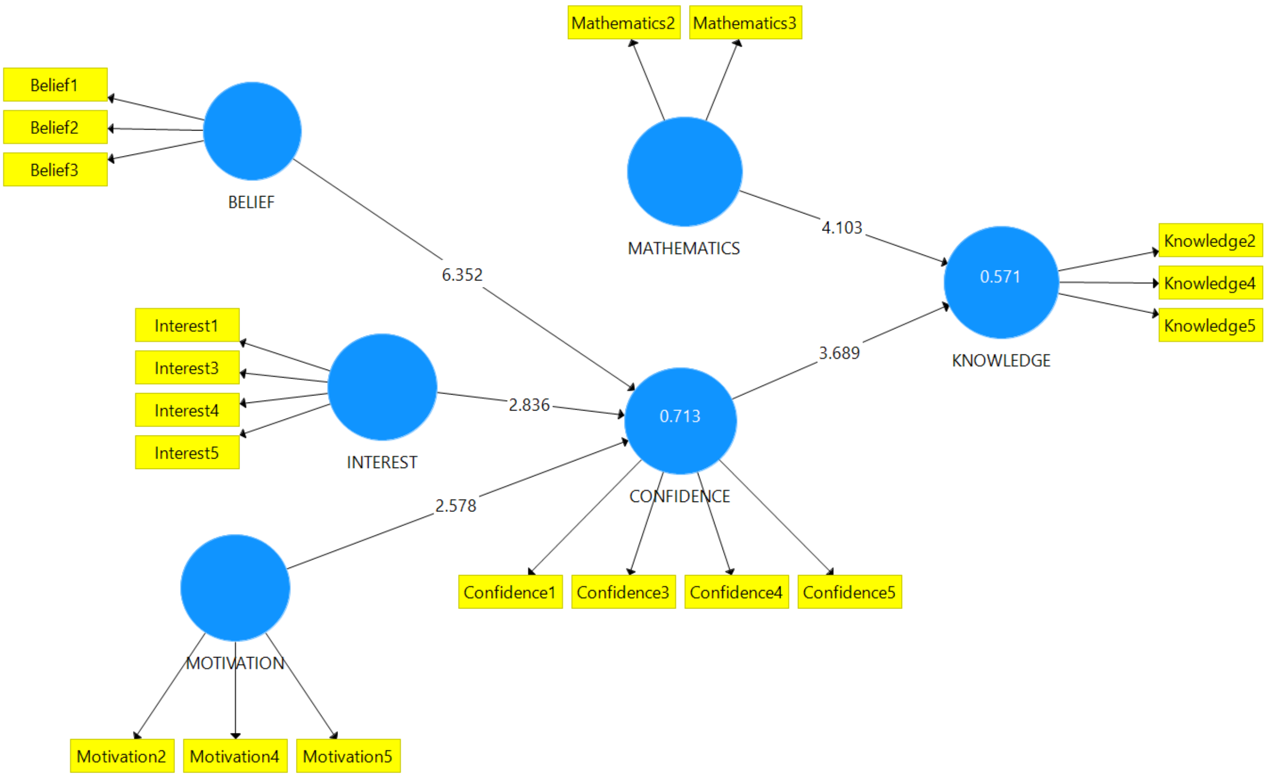

Fig. 4. R Square values (R^2) and inner model depicting t values.
(Note: a) LVs (constructs) show R square values and the inner model shows t values. b) Model result of bootstrapping subsample 5000.)

The R^2 values represent the amount of variance by the endogenous LVs (constructs) explained by all exogenous

LVs linked to it. As depicted in Table 12 and Fig. 4, the endogenous LV *Confidence* can be found to have a substantial to moderate outcome. Meanwhile, *Knowledge* can be expressed to have a moderate to substantial outcome [54, 64, 70]. Visual inspection of Fig. 4 shows that the endogenous LVs *Mathematics* and *Confidence* are Dependent Variables (DV) informed by the Independent Variables (IV). Note that while Confidence is regarded as endogenous, it is also a mediating variable that acts as a DV that informs the IV Knowledge. Further calculation of the effect strength (f^2) for each IV allows the assessment of the extent to which the IV contributes to explaining the DV.

5) Effect size (f^2)

The assessment of the effect size allows the observance of the effect of each exogenous LV on the endogenous LV. In doing so, the effect size assesses how strongly one exogenous LV contributes to explaining a certain endogenous LV in terms of R^2. The effect size formula is as follows:

$$f^2 = \frac{R^2_{included} - R^2_{excluded}}{1 - R^2_{included}}$$

$R^2_{included}$ and $R^2_{excluded}$ are the R^2 values of the endogenous latent variable (LV) when a selected exogenous LV is included or excluded from the model. The change in the R^2 values is calculated (Table 13) by estimating the PLS path model twice, that is, once with the exogenous LV included (yielding $R^2_{included}$) and the second time with the exogenous LV excluded (yielding $R^2_{excluded}$). Therefore, the effect size (f^2) evaluation determines whether the omitted LV has a substantive impact on the endogenous construct, also known as the effect size of the exogenous LV on the model. The assessment of the effect size follows Cohen's guidelines, which are 0.02, 0.15 and 0.35 for small, medium and large effects, respectively [73]. Effect size values less than 0.02 indicate that there is no effect.

Table 13. Effect size (f^2)

Predictor	Endogenous	R-SQ Included	R-SQ Excluded	Effect size (f^2)	Outcome
Belief	Confidence	0.713	0.590	0.429	*strong*
Confidence	Knowledge	0.571	0.494	0.179	*moderate*
Interest	Confidence	0.713	0.675	0.132	*weak*
Mathematics	Knowledge	0.571	0.467	0.242	*moderate*
Motivation	Confidence	0.713	0.690	0.080	*weak*

Note: a) All effect sizes are valid since no $f^2 \leq 0.02$. b) Effect size impact indicators are, according to Cohen [73], f^2 values: 0.35 (large/strong), 0.15 (medium/moderate), and 0.02 (small/weak).

Similar findings by McGill [6] suggest that the factors of motivation and interest in computer programming through using robotics do not influence students' overall motivation levels. Meanwhile, a study by Jaipal-Jamani and Angeli [74] found that the implementation of robotics intervention significantly improved a subject's belief (moderate to high effect size), as they gained confidence in their capacity to complete the assignment successfully and believed they possessed the necessary skills to do so.

6) Predictive relevance (Q^2)

In addition to evaluating the magnitude of the R^2 values as a criterion of prediction accuracy, researchers also examine predictive relevance (Q^2), also known as the Stone-Geisser Q^2 value [75, 76]. This measure is an indicator of the model's predictive power or predictive relevance. The Q^2 value is obtained through blindfolding procedures for a specified omission distance (D) with values between 5 and 10. Q^2 values larger than zero suggest that the model has predictive relevance for a certain endogenous LV [54, 70, 77]. As remarked by Garson [34], "values of Q^2 greater than 0 means that the PLS-SEM model is predictive of the given endogenous variable under scrutiny". In contrast, values of 0 and below indicate a lack of predictive relevance. Cohen [73] prescribes $0.02 \leq Q^2 < 0.15$: *small* effect size, $0.15 \leq Q^2 < 0.35$: *medium* effect size and $Q^2 \geq 0.35$: *high* effect size. Predictive relevance is assessed through the findings of the construct cross-validation redundancy, which addresses model fit [34].

Findings of the Q^2 values (Table 14) indicate that endogenous LVs *confidence* and *knowledge* possess high predictive relevance due to $Q^2 > 0.35$. Further, it suggests that the LVs that inform *confidence* and *knowledge* make a meaningful contribution to promoting predictive relevance and that the model is meaningful. Overall, the model depicts that belief in coding, interest in coding and motivation to code encourage an individual's confidence to learn code. These factors significantly influence the student's enthusiasm and self-awareness of their ability while learning how to code—knowledge/*outcome*. Thus, it supports the findings from Mason and Rich [78] that frequent coding positively influences attitudes such as beliefs, interests and motivation.

Table 14. Construct cross-validated redundancy

Items	SSO	SSE	Q^2(= 1-SSE/SSO)
Belief	225.000	225.000	
Confidence	300.000	189.747	0.368
Interest	300.000	300.000	
Knowledge	225.000	137.841	0.387
Mathematics	150.000	150.000	
Motivation	225.000	225.000	

Note: a) Blindfolding omission distance D= 7. b) SSO represents the mean value prediction. c) SSE is the prediction error when using the model prediction. d) All $Q^2 > 0$ showing model has predictive relevance [54, 70, 77]. e) Predictor Relevance (Q^2) of predictor exogenous latent variables as according to Henseler *et al.* [70], Q^2 values: 0.35 (large), 0.15 (medium), and 0.02 (small). f) All $Q^2 \geq 0.35$ high effect.

V. Conclusion

This study's objective was to investigate the factors contributing to one's *knowledge* of programming through using robotics by examining the influence of *confidence*, *interest*, *motivation*, *belief*, *Mathematics*, and *knowledge.* A survey to gauge 75 participants' afterthoughts enabled the testing of a conceptualised model using PLS-SEM analysis. The model met all respective criteria within the measurement and structural models, thereby deeming the model valid and successful in explaining students' perceptions of robotics when learning to program. All hypotheses were supported since $p < 0.05$ confirms that of the literature that relationships H1, H2, H3, H4 and H5 play a crucial role in learning computer programming. In addition, it can be inferred to some extent that using physical manipulatives such as robotics has a positive effect on learning text-based computer programming. Learning to code was highly engaging, which gave meaning to the code, in contrast to viewing the execution of the code on the screen in a 2D environment. This study paves the way for further research with larger datasets and differing contexts that may investigate other possible contributing factors that influence the learning of computer programming. The outcomes of this study have the potential to significantly contribute to the understanding of the adoption and utilisation of physical manipulatives in learning text-based computer programming in South Africa and similar contexts globally.

INFORMED CONSENT

Informed consent was gathered from all participating students. Confidentiality was maintained by not requesting names or any other information that would identify the students involved. The subjects were informed of their right to withdraw from the investigation at any time.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Conceptualization, R.G.G.; methodology, R.G.G.; validation and formal analysis, R.G.G.; investigation, R.G.G.; data curation, R.G.G.; writing original draft preparation, R.G.G.; writing review and editing, R.G.G. and D.W.G.; visualization, R.G.G.; project administration, R.G.G.; funding acquisition, R.G.G. Both authors had approved the final version.

FUNDING

The authors disclosed receipt of the following financial support for the research. This work was supported by the National Research Foundation awarded to RG [grant number: TTK2204092902].

ACKNOWLEDGEMENT

The authors would like to thank all participants for taking part in this study. The study emanates from RG's PhD project.

REFERENCES

[1] C. S. Cheah, "Factors contributing to the difficulties in teaching and learning of computer programming: A literature review," *Contemporary Educational Technology*, vol. 12, no. 2, ep272, 2020. https://doi.org/10.30935/cedtech/8247

[2] A. Alam, "Educational robotics and computer programming in early childhood education: A conceptual framework for assessing elementary school students' computational thinking for designing powerful educational scenarios," in *Proc. International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN)*, 2022, pp. 1–7. https://doi.org/10.1109/ICSTSN53084.2022.9761354

[3] R. G. Govender and D. W. Govender, "Using robotics in the learning of computer programming: Student experiences based on experiential learning cycles," *Education Science*, vol. 13, no. 3, pp. 1–13, 2023. https://doi.org/10.3390/educsci13030322

[4] Z. A. Kersteen, M. C. Linn, M. Clancy, and C. Hardyck, "Previous experience and the learning of computer programming: The computer helps those who help themselves," *Journal of Educational Computing Research*, vol. 4, no. 3, pp. 321–333, 1988. https://doi.org/10.2190/9LE6-MBXA-JDPG-UG90

[5] A. Robins, J. Rountree, and N. Rountree, "Learning and teaching programming: A review and discussion," *Computer Science Education*, vol. 13, no. 2, pp. 137–172, 2003. https://doi.org/10.1076/csed.13.2.137.14200

[6] M. M. McGill, "Learning to program with personal robots: Influences on student motivation," in *Proc. ACM Transactions on Computing Education (TOCE)*, vol. 12, no. 1, 2012, pp. 1–32. https://doi.org/10.1145/2133797.2133801

[7] R. G. Govender, "Embracing the fourth industrial revolution by developing a more relevant educational spectrum: Coding, robotics, and more," in *Teaching and Learning in the 21st Century*, J. Naidoo Ed., Brill, SENSE, 2021, ch. 3, pp. 30–49. https://doi.org/10.1163/9789004460386_003

[8] A. Sophokleous, P. Christodoulou, L. Doitsidis, and S. A. Chatzichristofis, "Computer vision meets educational robotics," *Electronics*, vol. 10, no. 6, p. 730, 2021. https://doi.org/10.3390/electronics10060730

[9] J. Noh and J. Lee, "Effects of robotics programming on the computational thinking and creativity of elementary school students," *Educational Technology Research and Development*, vol. 68, pp. 463–484, 2020. https://doi.org/10.1007/s11423-019-09708-w

[10] M. Çetin and H. Ö. Demircan, "Empowering technology and engineering for STEM education through programming robots: A systematic literature review," *Early Child Development and Care*, vol. 190, no. 9, pp. 1323–1335, 2020. https://doi.org/10.1080/03004430.2018.1534844

[11] O. Coşkunserçe, "Comparing the use of block-based and robot programming in introductory programming education: Effects on perceptions of programming self-efficacy," *Computer Applications in Engineering Education*, vol. 31, no. 5, pp. 1234–1255, 2023. https://doi.org/10.1002/cae.22637

[12] T. Sapounidis and D. Alimisis, "Educational robotics curricula: Current trends and shortcomings," in *Proc. Educational Robotics International Conference*, 2021, pp. 127–138. https://doi.org/10.1007/978-3-030-77022-8_12

[13] L. Fortunati, A. M. Manganelli, and G. Ferrin, "Arts and crafts robots or LEGO® MINDSTORMS robots? A comparative study in educational robotics," *International Journal of Technology and Design Education*, vol. 32, no. 1, pp. 287–310, 2022. https://doi.org/10.1007/s10798-020-09609-7

[14] A. Kovari and J. Katona, "Effect of software development course on programming self-efficacy," *Education and Information Technologies*, vol. 28, no. 1, pp. 1–27, 2023. https://doi.org/10.1007/s10639-023-11617-8

[15] T. Daradoumis, J. M. Marquès Puig, M. Arguedas and L. Calvet Liñan, "Enhancing students' beliefs regarding programming self-efficacy and intrinsic value of an online distributed programming environment," *Journal of Computing in Higher Education*, vol. 34, no. 3, pp. 577–607, 2022. https://doi.org/10.1007/s12528-022-09310-9

[16] J. H. Zhang, B. Meng, L. C. Zou, Y. Zhu and G. J. Hwang, "Progressive flowchart development scaffolding to improve university students' computational thinking and programming self-efficacy," *Interactive Learning Environments*, vol. 31, no. 6, pp. 3792–3809, 2023. https://doi.org/10.1080/10494820.2021.1943687

[17] J. Flegg, D. Mallet, and M. Lupton, "Students' perceptions of the relevance of mathematics in engineering," *International Journal of Mathematical Education in Science and Technology*, vol. 43, no. 6, pp. 717–732, 2012. https://doi.org/10.1080/0020739X.2011.644333

[18] G. Kaiser, "Mathematical modelling and applications in education," in *Encyclopedia of Mathematics Education*, S. Lerman, Ed., Springer, Cham, 2020. https://doi.org/10.1007/978-3-030-15789-0_101

[19] J. J. Tua and J. R. Johnson, "Can computer programming improve problem-solving ability?" *ACM SIGCSE Bulletin*, vol. 22, no. 2, pp. 30–33, 1990. Available: https://doi.org/10.1145/126445.126451

[20] D. W. Dalton and D. A. Goodrum, "The effects of computer programming on problem-solving skills and attitudes," *Journal of Educational Computing Research*, vol. 7, no. 4, pp. 483–506, 1991. https://doi.org/10.2190/762V-KV6T-D3D1-KDY2

[21] H. Jiang, D. Turnbull, X. Wang, R. Chugh, Y. Dou, and S. Chen, "How do mathematics interest and self-efficacy influence coding interest and self-efficacy? A structural equation modeling analysis," *International Journal of Educational Research*, vol. 115, no. 1, 102058, 2022. https://doi.org/10.1016/j.ijer.2022.102058

[22] P. J. Rich, N. Bly, and K. R. Leatham, "Beyond cognitive increase: Investigating the influence of computer programming on perception and application of mathematical skills," *Journal of Computers in Mathematics and Science Teaching*, vol. 33, no.1, pp. 103–128, 2014.

[23] S. Bergin and R. Reilly, "Programming: Factors that influence success," *ACM SIGCSE Bulletin*, vol. 37, no. 1, pp. 411–415, 2005. https://doi.org/10.1145/1047124.1047480

[24] S. Milner. (1973). The effects of computer programming on performance in Mathematics. in *Proc. Annual Meeting of the American Educational Research Association (AERA)*. New Orleans, American Educational Research Association. [Online]. Available: https://files.eric.ed.gov/fulltext/ED076391.pdf

[25] S. Psycharis and M. Kallia, "The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy

and problem-solving," *Instructional Science*, vol. 45, no. 5, pp. 583–602, 2017. https://doi.org/10.1007/s11251-017-9421-5

[26] O. T. Kaufmann and B. Stenseth, "Programming in mathematics education," *International Journal* of *Mathematical Education in Science and Technology*, vol. 52, no. 7, pp. 1029–1048, 2021. https://doi.org/10.1080/0020739X.2020.1736349

[27] M. J. Tsai, C. Y. Wang, and P. F. Hsu, "Developing the computer programming self-efficacy scale for computer literacy education," *Journal of Educational Computing Research*, vol. 56, no. 8, pp. 1345–1360, 2019. https://doi.org/10.1177/0735633117746747

[28] A. Carbone, J. Hurst, I. Mitchell, and D. Gunstone, "An exploration of internal factors influencing student learning of programming," in *Proc. the Eleventh Australasian Conference on Computing Education*, vol. 95, Australian Computer Society, 2009, pp. 25–34.

[29] J. Blanchard, "Hybrid environments: A bridge from blocks to text," in *Proc. the 2017 ACM Conference on International Computing Education Research*, 2017, pp. 295–296. https://doi.org/10.1145/3105726.3105743

[30] M. Biggers, A. Brauer, and T. Yilmaz, "Student perceptions of computer science: A retention study comparing graduating seniors with cs leavers," *ACM SIGCSE Bulletin*, vol. 40, no. 1, pp. 402–406, 2008. Available: https://doi.org/10.1145/1352322.1352274

[31] P. Byrne and G. Lyons, "The effect of student attributes on success in programming," *ACM SIGCSE Bulletin*, vol. 33, no. 3, pp. 49–52, 2001. https://doi.org/10.1145/507758.377467

[32] T. Daly, "Minimizing to maximize: An initial attempt at teaching introductory programming using Alice," *Journal of Computing Sciences in Colleges*, vol. 26, no. 5, pp. 23–30, 2011.

[33] J. F. Hair, G. T. M. Hult, C. M. Ringle, M. Sarstedt, N. P. Danks, S. Ray, "Evaluation of reflective measurement models," *Partial Least Squares Structural Equation Modeling (PLS-SEM) Using R. Classroom Companion: Business*, Springer, Cham, 2021. https://doi.org/10.1007/978-3-030-80519-7_4

[34] G. D. Garson, *Partial Least Squares: Regression and Structural Equation Models*, Statistical Associates Publishers, 2016.

[35] J. F. Hair, G. T. M. Hult, C. Ringle, and M. Sarstedt, *A Primer on Partial Least Squares Structural Equation Modeling (PLS-SEM)*, Sage, 2016.

[36] M. E. Douglas, S. Peecksen, J. Rogers, and M. Simmons, "College students' motivation and confidence for ePortfolio use," *International Journal of ePortfolio*, vol. 9, no. 1, pp. 1–16, 2019.

[37] K. A. Blotnicky, T. Franz-Odendaal, F. French and P. Joy, "A study of the correlation between STEM career knowledge, mathematics self-efficacy, career interests, and career activities on the likelihood of pursuing a STEM career among middle school students," *International Journal of STEM Education*, vol. 5, no. 1, pp. 1–15, 2018. https://doi.org/10.1186/s40594-018-0118-3

[38] M. Fırat, H. Kılınç, and T. V. Yüzer, "Level of intrinsic motivation of distance education students in e-learning environments," *Journal of Computer Assisted Learning*, vol. 34, no. 1, pp. 63–70, 2018. https://doi.org/10.1111/jcal.12214

[39] M. LaForce, E. Noble, and C. Blackwell, "Problem-Based Learning (PBL) and student interest in STEM careers: The roles of motivation and ability beliefs," *Education Sciences*, vol. 7, no. 4, 92, 2017. https://doi.org/10.3390/educsci7040092

[40] M. Fischer and D. Sliwka, "Confidence in knowledge or confidence in the ability to learn: An experiment on the causal effects of beliefs on motivation," *Games and Economic Behavior*, vol. 111, no. 1, pp. 122–142, 2018. https://doi.org/10.1016/j.geb.2018.02.005

[41] I. Sadler, "The role of self-confidence in learning to teach in higher education," *Innovations in Education and Teaching International*, vol. 50, no. 2, pp. 157–166, 2013. https://doi.org/10.1080/14703297.2012.760777

[42] J. F. Hair, M. Sarstedt, C. M. Ringle, and J. A. Mena, "An assessment of the use of partial least squares structural equation modeling in marketing research," *Journal of the Academy of Marketing Science*, vol. 40, no. 3, pp. 414–433, 2012. https://doi.org/10.1007/s11747-011-0261-6

[43] L. Law and N. Fong, "Applying partial least squares structural equation modeling (PLS-SEM) in an investigation of undergraduate students' learning transfer of academic English," *Journal of English for Academic Purposes*, vol. 46, pp. 1–22, 2020. https://doi.org/10.1016/j.jeap.2020.100884

[44] L. Lin, Z. Huang, B. Othman, and Y. Luo, "Let's make it better: An updated model interpreting international student satisfaction in China based on PLS-SEM approach," *Plos One*, vol. 15, no. 7, e0233546, 2020. https://doi.org/10.1371/journal.pone.0233546

[45] N. A. Ramli, H. Latan, and G. V. Nartea, "Why should PLS-SEM be used rather than regression? Evidence from the capital structure perspective," in *Partial Least Squares Structural Equation Modeling*, K. N. Avkiran and C. M. Ringle, Eds., Springer, 2018, pp. 171–209.

[46] J. Hulland, "Use of Partial Least Squares (PLS) in strategic management research: A review of four recent studies," *Strategic Management Journal*, vol. 20, no. 2, pp. 195–204, 1999. https://doi.org/10.1002/(SICI)1097-0266(199902)20:2<195::AID-SMJ13>3.0.CO;2-7

[47] J. F. Hair, M. Sarstedt, L. Hopkins, and V. G. Kuppelwieser, "Partial least squares structural equation modeling (PLS-SEM): An emerging tool in business research," *European Business Review*, vol. 26, no. 2, pp. 106–121, 2014. https://doi.org/10.1108/EBR-10-2013-0128

[48] A. H. Memon and I. A. Rahman, "SEM-PLS analysis of inhibiting factors of cost performance for large construction projects in Malaysia: Perspective of clients and consultants," *The Scientific World Journal*, vol. 2014, no. 1, 165158, 2014. https://doi.org/10.1155/2014/165158

[49] J. C. Nunnally, *Psychometric Theory*, 2nd ed., McGraw-Hill, 1978.

[50] T. K. Dijkstra and J. Henseler, "Consistent partial least squares path modeling," *MIS Quarterly*, vol. 39, no. 2, pp. 297–316, 2015. https://doi.org/10.25300/MISQ/2015/39.2.02

[51] R. P. Bagozzi and Y. Yi, "On the evaluation of structural equation models," *Journal of the Academy of Marketing Science*, vol. 16, no. 1, pp. 74–94, 1988. https://doi.org/10.1007/BF02723327

[52] C. Fornell and D. F. Larcker," Evaluating structural equation models with unobservable variables and measurement error," *Journal of Marketing Research*, vol. 18, no. 1, pp. 39–50, 1981. https://doi.org/10.2307/3151312

[53] D. Gefen, D. Straub, and M. C. Boudreau, "Structural equation modeling and regression: Guidelines for research practice," *Communications of the Association for Information Systems*, vol. 4, no. 1, pp. 1–78, 2000. https://doi.org/10.17705/1CAIS.00407

[54] W. W. Chin, "Commentary: issues and opinions on structural equation modelling," *MIS Quarterly*, vol. 22, no. 1, pp. 7–16, 1998. http://www.jstor.org/stable/249674

[55] G. Demo, E. R. Neiva, I. Nunes, and K. Rozzett, "Human resources management policies and practices scale (HRMPPS): Exploratory and confirmatory factor analysis," *BAR-Brazilian Administration Review*, vol. 9, no. 4, pp. 395–420, 2012. https://doi.org/10.1590/S1807-76922012005000006

[56] W. W. Chin, "How to write up and report PLS analyses," in *Handbook of Partial Least Squares*, V. E. Vinzi, W. W. Chin, J. Henseler, and H. Wang, Eds., Springer, 2010, pp. 655–690.

[57] J. A. Godwin. (2021). Linear regression with bootstrapping. *Towards Data Science*. [Online]. Available: https://towardsdatascience.com/linear-regression-with-bootstrapping-4924c05d2a9#:~:text=The%20bootstrap%20method%20can%20be,small%20changes%20in%20data%20values

[58] L. T. DeCarlo, "On the meaning and use of kurtosis," *Psychological Methods*, vol. 2, no. 3, pp. 292–307, 1997. https://doi.org/10.1037/1082-989X.2.3.292

[59] K. V. Mardia, "Measures of multivariate skewness and kurtosis with applications," *Biometrika*, vol. 57, no. 1, pp. 519–530, 1970. https://doi.org/10.2307/2334770

[60] K. V. Mardia, "Applications of some measures of multivariate skewness and kurtosis in testing normality and robustness studies," *Sankhya: The Indian Journal of Statistics*, vol. 36, no. 2, pp. 115–128, 1974.

[61] J. Frost. Introduction to bootstrapping in statistics with an example. *Making Statistics Intuitive*. [Online]. Available: https://statisticsbyjim.com/hypothesis-testing/bootstrapping/

[62] J. Henseler, C. M. Ringle, and M. Sarstedt, "A new criterion for assessing discriminant validity in variance-based structural equation modeling, *Journal of the Academy of Marketing Science*, vol. 43, no. 1, pp. 115–135, 2015. https://doi.org/10.1007/s11747-014-0403-8

[63] C. M. Ringle. (2015). HTMT discriminant validity. *SmartPLS*. [Online]. Available: https://forum.smartpls.com/viewtopic.php?t=3616

[64] J. F. Hair, C. M. Ringle, and M. Sarstedt, "PLS-SEM: Indeed, a silver bullet," *Journal of Marketing Theory and Practice*, vol. 19, no. 2, pp.139–152, 2011. https://doi.org/10.2753/MTP1069-6679190202

[65] A. Diamantopoulos and J. A. Siguaw, "Formative versus reflective indicators in organizational measure development: Comparison and empirical illustration," *British Journal of Management*, vol. 17, no. 4, pp. 263–282, 2006. https://doi.org/10.1111/j.1467-8551.2006.00500.x

[66] D. X. Peng, and F. Lai, "Using partial least squares in operations management research: A practical guideline and summary of past research," *Journal of Operations Management*, vol. 30, no. 6, pp. 467–480, 2012. https://doi.org/10.1016/j.jom.2012.06.002

[67] S. Hadad, T. Shamir-Inbal, I. Blau, and E. Leykin, "Professional development of code and robotics teachers through Small Private Online Course (SPOC): Teacher centrality and pedagogical strategies

for developing computational thinking of students," *Journal of Educational Computing Research*, vol. 59, no. 4, pp. 763–791, 2021. https://doi.org/10.1177/0735633120973432

[68] M. E. Karim, S. Lemaignan, and F. Mondada, "A review: Can robots reshape K-12 STEM education?" in *Proc. 2015 IEEE International Workshop on Advanced Robotics and its Social Impacts (ARSO)*, Curran Associates, 2016, pp. 1–8. https://doi.org/10.1109/ARSO.2015.7428217

[69] M. Zhang and Y. Wan, "Improving learning experiences using LEGO Mindstorms EV3 robots in control systems course," *The International Journal of Electrical Engineering & Education*, pp. 1–23, 2020. https://doi.org/10.1177/0020720920965873

[70] J. Henseler, C. M. Ringle, and R. R. Sinkovics, "The use of partial least squares path modeling in international marketing," in *New challenges to International Marketing (Advances in International Marketing)*, R. R. Sinkovics and P. N. Ghauri, Eds., Emerald Group Publishing, 2009, pp. 277–319.

[71] J. Karch and D. Van Ravenzwaaij, "Improving on adjusted R-squared," *Collabra: Psychology*, vol. 6, no. 1, 2020. https://doi.org/10.1525/collabra.343

[72] O. Harel, "The estimation of R 2 and adjusted R 2 in incomplete data sets using multiple imputation," *Journal of Applied Statistics*, vol. 36, no. 10, pp. 1109–1118, 2009. https://doi.org/10.1080/02664760802553000

[73] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, Lawrence Erlbaum, 1988.

[74] K. Jaipal-Jamani and C. Angeli, "Effect of robotics on elementary preservice teachers' self-efficacy, science learning, and computational thinking," *Journal of Science Education and Technology*, vol. 26, no. 1, pp. 175–192, 2017. https://doi.org/10.1007/s10956-016-9663-z

[75] S. Geisser, "A predictive approach to the random effect model," *Biometrika*, vol. 61, no. 1, pp. 101–107, 1974. https://doi.org/10.1093/BIOMET/61.1.101

[76] M. Stone, "Cross-validation and multinomial prediction," *Biometrika*, vol. 61, no. 3, pp. 509–515, 1974. https://doi.org/10.2307/2334733

[77] M. Tenenhaus, V. E. Vinzi, Y. M. Chatelin, and C. Lauro, "PLS path modeling," *Computational Statistics & Data Analysis*, vol. 48, no. 1, pp. 159–205, 2005. https://doi.org/10.1016/j.csda.2004.03.005

[78] S. L. Mason and P. J. Rich, "Development and analysis of the elementary student coding attitudes survey," *Computers & Education*, vol. 153, no. 1, 103898, 2020. https://doi.org/10.1016/j.compedu.2020.103898

Copyright © 2025 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (CC BY 4.0).