Investigating the Attitudes and Emotions of K-12 Students Towards Debugging

Laurie Gale
Raspberry Pi Computing Education Research Centre
University of Cambridge
Cambridge, UK
lpg28@cam.ac.uk

Sue Sentance
Raspberry Pi Computing Education Research Centre
University of Cambridge
Cambridge, UK
ss2600@cam.ac.uk

ABSTRACT

Learning to program is a challenging process, known to instill a range of thoughts and feelings among learners. In particular, debugging is known to evoke emotional reactions in learners who struggle with it. While attitudes and emotions towards programming have previously been investigated, few studies are focused at the K-12 level, with even less specifically investigating the important skill of debugging. This paper reports on an exploratory study measuring the attitudes and emotions of K-12 students related to debugging. 73 students debugged five erroneous Python programs and answered questions on their perceived performance, attitudes, emotions, and debugging strategies. Analysis of students' survey responses revealed self-efficacy in debugging to be strongly correlated with gender, perceived performance, usefulness, and feelings of anxiety, with other associations also present. These findings contribute to our growing understanding of the challenges young people face when solving errors in computer programs.

CCS CONCEPTS

• **Social and professional topics → K-12 education**.

KEYWORDS

debugging, K-12 education, self-efficacy, computing education, programming education

ACM Reference Format:
Laurie Gale and Sue Sentance. 2023. Investigating the Attitudes and Emotions of K-12 Students Towards Debugging. In *The United Kingdom and Ireland Computing Education Research (UKICER) conference (UKICER 2023), September 7–8, 2023, Swansea, Wales Uk.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3610969.3611120

1 INTRODUCTION

Debugging, the process of finding and fixing errors in a program, is a skill of critical importance in programming. Not only is debugging important to master to become a proficient programmer, it is also a component of computational thinking [14]. Furthermore, debugging can be considered an instance of troubleshooting [17, 21], which some claim can be transferred to other subject domains [6].

Despite its utility, debugging is generally a difficult process for students to learn. This is made more difficult for secondary students learning their first text-based programming language, whose computing teacher may not be confident teaching programming [31, 37] or feel challenged by having to provide troubleshooting support to many students in the classroom [28]. Students' difficulty with solving errors may lead to emotional reactions [18] which in turn can induce negative attitudes towards debugging [22]. If these attitudes persist, the risk that young learners become disinterested in programming arises.

It is important to capture attitudes towards debugging specifically in order to improve how it is taught and perceived. To the authors' knowledge, attitudes and emotions towards debugging have not been investigated at a lower secondary level, even though many students around the world are now learning to program in a text-based language.

This paper reports on part of a study that investigates the attitudes and behaviours of lower secondary students towards debugging. Students were presented with erroneous Python programs that they attempted to debug then completed an online survey relating to their attitudes and emotions towards debugging. Exploratory analysis of the survey responses is then reported and discussed. The research questions for this paper are as follows:

- RQ1: What are the attitudinal factors (self-efficacy, usefulness, and general perceptions) of lower secondary students towards debugging?
- RQ2: What emotions (anxiety, frustration, and joy) do lower secondary students feel when debugging their code?

As well as furthering our understanding of the relations of attitudes and emotions towards debugging, the results of this study can be used by teachers to adjust programming teaching practices such that the attitudinal and emotional challenges of debugging are tackled. This holds particular promise for less confident students.

2 RELATED WORK

An attitude can be defined as an evaluative judgment about a particular phenomenon [24]. This is a complex concept containing many interrelated components, such as the perceived usefulness of a subject and self-efficacy, one's belief that one can achieve an outcome by successfully completing relevant tasks [1]. A learner's first experiences with a subject or skill are a significant determinant of the attitudes they go on to form [22]. In particular, self-efficacy is informed by the success and emotional load of these initial experiences [1]. This is no different for learning to program, where such experiences can be emotionally taxing if a learner finds the content difficult to grasp [13, 28].

UKICER 2023, September 7–8, 2023, Swansea, Wales Uk

Attitudes and emotions specifically towards debugging in K-12 students are not well investigated. Chen et al. [7] studied the debugging behaviour and attitudes of 72 grade 12 students, with questionnaire responses revealing a general belief that debugging proficiency was based on an individual's ability and could not be learnt. When interviewing 12 high school teachers about their experiences of teaching debugging, Michaeli and Romeike [28] found mentions of the feeling of helplessness that arose in 'weaker' students when encountering an error. Frustration in students struggling to debug was also commonly referenced, often associated with requesting support from the teacher. Emotional reactions to debugging were also mentioned by five of the 12 undergraduate students interviewed in Whalley et al. [36], with roughly half of these mentions being negative reactions.

Studies that focus on attitudes and emotions towards programming generally are more common. Kinnunen and Simon [18] interviewed nine CS1 students about their emotional experiences with assignments on their course. The frustration of being unable to solve errors was associated with negative feelings such as despair and inadequacy, with some students referring to a 'hit by lightning' experience when unexpectedly encountering an error. These findings further highlight the link between early programming experiences and strong emotions.

Building on Kinnunen and Simon [18], Lishinski et al. [23] created a four-question survey to measure the emotions felt by undergraduates over the duration of a CS1 course. Using path analysis, feelings of frustration were found to have both a short- and long-term effect on students' performance, with self-efficacy and inadequacy also having longer-term impacts. Significant gender differences were present, with girls reporting lower self-efficacy and more feelings of frustration and anxiety [23]. However, the survey was not externally validated and only contained four questions, meaning students' actual emotions during the course may not have been accurately captured.

Electrodermal activity, a measure of sweat production and a proxy for emotional response, has also been used to investigate emotions experienced when programming. Gorson et al. [13] used this measure on CS1 undergraduates completing programming assignments, identifying 21 emotional triggers. Interestingly, two of the five most cited triggers of negative emotions were both encountering and struggling to solve an error. Successful debugging was mentioned as a source of positive emotion but was not referenced as much as its negative counterparts. Despite not focusing on debugging in particular, these studies still highlight the emotional angst of solving errors for introductory programmers.

A summary of theoretical constructs on attitudes and emotions developed in programming education is provided by Malmi et al. [25]. Of the 50 constructs reviewed, 21 involved self-efficacy, 11 were instruments for measuring attitudes or emotions, but none were specifically related to debugging, despite its emotional nature.

Prior literature has highlighted how debugging is an emotional process, which has a significant influence on one's experiences of learning programming. If a learner's initial experiences with solving errors are frustrating and overwhelming [13, 18, 28], their attitudes towards programming are likely to be negative [22]. This in turn reduces the likelihood that a student will enjoy or engage with programming, or perhaps computing, in the future.

3 METHOD

The study consisted of two main phases. Participants were first presented with five debugging exercises containing erroneous Python programs and then completed a survey relating to their attitudes and emotions towards debugging.

3.1 Participants

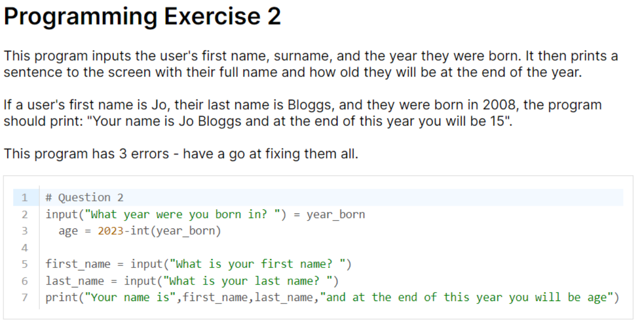
Computing teachers at schools local to the authors were invited to conduct the study in their classrooms, of which three (from two state-funded schools) accepted. In total, 75 lower secondary students aged 12-14 (grades 7 and 8) participated. As two students did not complete the study, the results of 73 students were analysed.

Students in each class were of a range of abilities and had been learning Python for a few months to a year, with prior experience with block-based languages. Gender was self-reported at the end of the survey, with 36 males, 24 females, 3 reporting as other, and 10 preferring not to disclose their gender.

Approval to conduct the study was granted by the Department of Computer Science and Technology ethics committee at the University of Cambridge. Before participating, consent of the student, a parent or guardian, and their computing teacher was obtained.

3.2 Procedure and Data Collection

Students began the study by attempting five debugging exercises in a web-based environment, an example of which is shown in Figure 1. Each exercise consisted of a Python program containing several errors, a description of the program's expected behaviour, and the number of errors in the program. The authors define debugging as 'the process of finding and fixing errors in a computer program', where errors may be syntactical or semantic. As a result, the programs contained a range of syntax, runtime, and logical errors commonly found in novices' programs [19, 30, 33, 34]. The exercises increased in difficulty and the number of errors. Additionally, edits made by the students to the programs were logged and stored to analyse the patterns that students used to debug their programs. These results will be reported in a future study.



Programming Exercise 2

This program inputs the user's first name, surname, and the year they were born. It then prints a sentence to the screen with their full name and how old they will be at the end of the year.

If a user's first name is Jo, their last name is Bloggs, and they were born in 2008, the program should print: "Your name is Jo Bloggs and at the end of this year you will be 15".

This program has 3 errors - have a go at fixing them all.

```
# Question 2
input("What year were you born in? ") = year_born
  age = 2023-int(year_born)

first_name = input("What is your first name? ")
last_name = input("What is your last name? ")
print("Your name is",first_name,last_name,"and at the end of this year you will be age")
```

Figure 1: A Programming Exercise Used in the Study

Students then completed an online survey containing statements relating to attitudes and emotions that are typically associated with debugging, responding on a five-point Likert scale. Two open-ended questions relating to the students' perceived performance and debugging techniques employed were also included, as well as demographic questions on the student's school, year group, and gender. This survey was developed by the authors and partially based on other validated surveys relating to computing [11, 32], although the difference in granularity between debugging as a skill and computing as a subject meant that some questions specifically related to debugging were needed. Using a non-standardised instrument allowed for more flexibility in the data collected, particularly appropriate for exploratory research [26]. Two questions per attitude and emotion (henceforth referred to as 'construct') were included; the survey had to be short enough to keep lower secondary students engaged and avoid spoiled responses, especially as they would already have attempted the potentially challenging debugging exercises prior to completing the survey.

3.3 Data Analysis

Due to the exploratory nature of the study, no hypotheses were formulated, nor were there variables to compare the values of over time. Rather, the study's focus was on general patterns in the students' survey responses, which was done through the methods of analysis detailed below. The statistical software package Stata was used to store, structure, and perform the analysis.

For each statement with a Likert response, the skewness and median response were calculated, with the median selected as the preferred measure of central tendency for ordinal data [15].

Responses to the open-ended questions were coded and analysed using qualitative content analysis [27] in the QDA software NVivo. An inductive approach was used to iteratively code the data and generate a set of themes representing students' comments on their performance and debugging strategies that they employed. The second author also coded the responses using the codebook devised by the first author, with an interrater reliability rating of $\kappa = 0.79$, indicating substantial agreement [16].

A correlation matrix was generated to identify associations between survey items with a Likert response. This was done by pairing survey items measuring the same construct, where the pair of items had a Cronbach's alpha value of above 0.7, as this suggests reliable items [8]. General perceptions and joy were not paired as a result. Spearman's rho was the chosen correlation measure as this is commonly used for ordinal data [8].

Chi-square tests of independence were conducted between each survey construct (including perceived performance) and the main themes from the content analysis to explore whether any themes mentioned in the free-text questions were related to attitudes and emotions towards debugging. Themes from the free-text responses were treated as binary variables based on whether a student mentioned them.

Hierarchical agglomerative cluster analysis with average linkage was conducted to identify similar groups of responses among participants. However, the clusters generated did not provide additional insight to the correlation matrix, so these results are not reported.

3.4 Validity and Reliability

Feedback from the survey was obtained from several computing education researchers to ensure the questions were worded appropriately and measured what they intended to measure. One of the teachers partaking in the study also provided feedback for the debugging exercises and survey, which was acted upon to ensure the study material was appropriate for lower secondary students.

The internal consistency of all 13 statements with a Likert scale response was calculated using Cronbach's alpha, resulting in a value of $\alpha = 0.806$, indicating highly reliable items [8]. To correct for the multiple correlations calculated between the survey constructs, p-values were adjusted using the Benjamini-Hochberg procedure [3, 39]. Different correction methods, such as Bonferroni correction, were considered but not used due to concerns of being too conservative given the size of the correlation matrix [35].

4 RESULTS

Of the 13 survey items with Likert responses (see Table 1), 7 contained a non-neutral median response, with some of these questions illustrating skewed distributions. Statements 7, 9, and 11 each had a median response of 'Disagree'. Statements 7 and 11 had a $|skewness|$ of above 0.5, indicating a non-symmetrical distribution. On the contrary, statements 4, 6, and 12 had both negatively skewed distributions and a median response of 'Agree'. Statement 12, a statement regarding the enjoyment of debugging, had a skewness of -1.111, indicating a highly skewed distribution.

4.1 Correlation Analysis

4.1.1 Correlation of Emotions and Attitudes. The correlation matrix for the constructs and questions in the survey is displayed in Figure 2. Effect sizes are reported as per Cohen's [9] effect sizes for Spearman's rho. Note 'perceived performance' refers to the first question on the survey, which students answered on a scale of 1-5.

The matrix illustrates several significant correlations of varying strengths between the constructs. Notably, students' perceived performance and self-efficacy had statistically significant correlations with all other paired constructs aside from frustration. The strongest correlation ($r_s = 0.872, p < .001$) was between students' perceived performance on the debugging exercises and their self-efficacy in debugging, with a large effect size. The strongest negative correlation ($r_s = -0.490, p < .001$) existed between feelings of anxiety and perceived performance, with self-efficacy having an almost identical relationship ($r_s = -0.481, p < .001$). Both of these correlations had moderate effect sizes, indicating that students who reported higher levels of anxiety tended to report lower levels of perceived performance and self-efficacy.

Feelings of frustration and anxiety when debugging were also positively correlated with a large effect size ($r_s = 0.684, p =< 0.001$), indicating that the more frustrated a student feels when debugging, the more anxious they feel. In contrast, very little correlation is present between all other pairs of constructs.

4.1.2 Gender Correlations. Analysis of the subset of students reporting their gender as either male or female was used ($n = 60$) to explore correlations between gender and the constructs in the survey. The resulting correlations are displayed in Figure 3. The Benjamini-Hochberg procedure was again used to adjust the p-values associated with the elements in the matrix.

The strongest correlation involving gender was a negative correlation with self-efficacy with a moderate effect size ($r_s = -0.417, p = 0.009$). In other words, girls reported having lower levels of self-efficacy in debugging than boys. A negative correlation of medium

Table 1: Constructs and Statements Included in the Survey

#	Construct	Statement	Median Response	Skewness
1	Perceived Performance	"How well do you feel you performed when solving the errors in the programming exercises?"	OK	-0.058
2	Self-efficacy Q1	"When I get an error in a program, I am confident that I can solve it."	Neither agree nor disagree	-0.127
3	Self-efficacy Q2	"I am good at solving errors in computer programs."	Neither agree nor disagree	0.031
4	Usefulness Q1	"Being able to solve errors in computer programs helps me to solve problems in other subjects."	Agree	-0.261
5	Usefulness Q2	"Knowing how to solve errors in a computer program will help me later in life."	Neither agree nor disagree	-0.653
6	General Perceptions Q1	"To be good at debugging, you need to be good at programming."	Agree	-0.542
7	General Perceptions Q2	"Errors in programs should be rare if you are good at programming."	Disagree	0.668
8	Anxiety Q1	"When I have to fix an error in a program, I feel anxious."	Neither agree nor disagree	0.335
9	Anxiety Q2	"I feel afraid to debug a program as I'm worried I might add more errors."	Disagree	0.326
10	Frustration Q1	"When I am struggling to solve an error in a program, I get frustrated."	Neither agree nor disagree	-0.273
11	Frustration Q2	"Having to debug errors in a program makes me angry."	Disagree	0.660
12	Joy Q1	"When I solve an error, I feel happy with myself."	Agree	-1.111
13	Joy Q2	"I enjoy solving errors when I am programming."	Neither agree nor disagree	-0.088

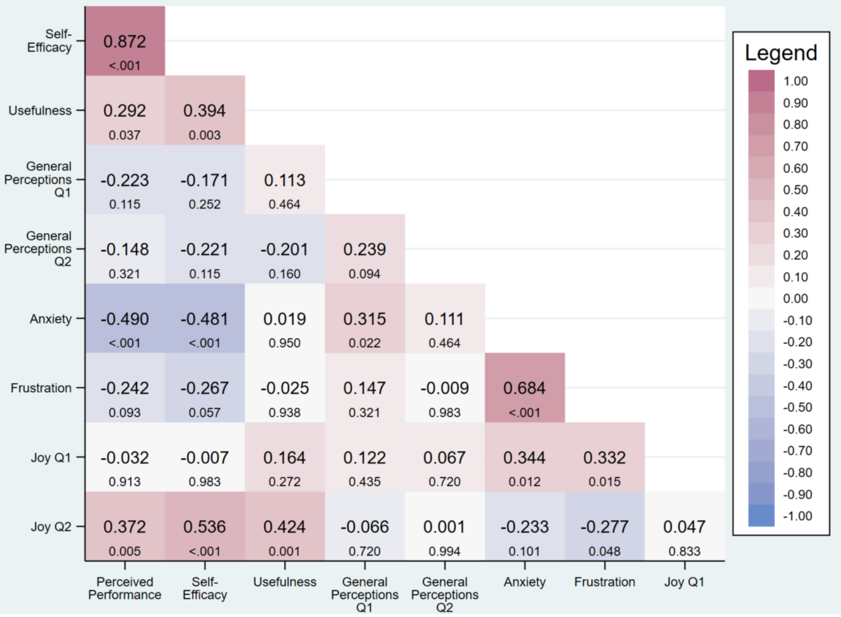


Figure 2: Correlation Matrix Showing Statements and Consistent Constructs (Key for Boxes: Top Value: r_s, Bottom Value: p)

effect size ($r_s = -.340, p = .027$) also existed between gender and statement (13), but the two questions on joy did not present a sufficient Cronbach's alpha value to be combined.

Although not statistically significant, other correlations of interest include a negative correlation with perceived performance ($r_s = -0.291, p = .064$) and a positive correlation with feelings of anxiety ($r_s = 0.283, p = .071$), both with small effect sizes. That is, girls tended to report more feelings of anxiety when debugging than boys did and vice versa for perceived performance.

4.2 Free-Text Responses

Tables 2 and 3 show the codebooks for the free-text questions in the survey. Here the strategies self-reported by students are described.

Students often referenced the use of multiple techniques for debugging their code, some of which indicated effective or ineffective

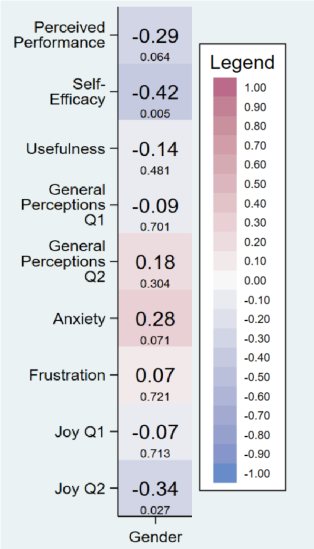


Figure 3: Correlation Matrix Including Gender (n = 60, 1 represents male and 2 represents female, Key for Boxes: Top Value: r_s, Bottom Value: p)

debugging behaviour. The most commonly mentioned strategy was the running of the code, usually at the start of the exercise for the purpose of obtaining the error message the code editor presented (34 mentions).

> *"I ran the code and looked at the error messages then focused on those lines to find and fix the errors."*

The repeated running of code was a strategy mentioned 10 times, sometimes with the implication that no or little changes were made between the runs, suggesting a misunderstanding of the deterministic nature of imperative programming.

> *"I ran the code multiple times to pick up any syntax errors."*

Other frequently mentioned strategies included the inspection of code at varying levels of granularity. Some students made the effort to inspect every line of code, while others seemed to generally inspect the whole program to see if anything 'stood out.'

> *"I looked for out of the normal pieces of code."*

On the other hand, some performed more targeted inspections of particular lines of code, sometimes informed by the error message.

> *"Check the lines of code where a bug is more likely first."*

Similar to these non-systematic methods, trial and error was mentioned by 10 students with a range of perceived performances. Some students were also able to debug their code by relying on their own programming knowledge, though these responses were often vague in describing how this knowledge was applied.

Finally, external resources proved favourable for some students, which mainly comprised of online sites (five mentions), a 'cheat sheet' (three mentions), or a 'knowledge organiser' (two mentions), the latter two likely provided by their teacher in previous lessons.

Table 2: Codebook for Responses to the Question "Why do you feel you performed this well?"

Name	Count
Progress on debugging exercises	37
Indicators of positive debugging behaviour	35
Null/low-information responses	23
Self-efficacy-related reasons	11
General performance on debugging exercises	10
Mention of programming knowledge	4
Time spent on debugging exercises	3
Missing small details	2
Trying hard	2

Table 3: Codebook for Responses to the Question "What techniques did you use to find and fix the errors in the programming exercises?"

Name	Count
Running of code	57
Inspection of code	25
Null/low-information responses	23
Use of external resources	12
Trial and error	10
Use of personal programming knowledge	6
Evidence of testing	2

4.3 Mapping of Free-Text Responses to Attitudes

Several relations between the themes in Tables 2 and 3 and the survey constructs were found. In terms of debugging strategies, the mention of 'running of code' was found to be associated with perceived performance ($\chi^2(1) = 18.08, p = .001$) and self-efficacy ($\chi^2(1) = 27.70, p = .001$). Students who mentioned running code, at the beginning of the exercise or otherwise, generally answered more positively towards the questions on perceived performance and self-efficacy compared with those who did not mention such a strategy. Interestingly, the use of external resources was found to be related to self-efficacy ($\chi^2(1) = 15.94, p = .043$) and anxiety ($\chi^2(1) = 17.37, p = .027$). Students who explicitly referred to consulting external resources generally felt less confident and more anxious debugging compared to those who did not mention this strategy. No significant associations for other strategies were present, perhaps due to the small number of students who cited them.

When describing how students felt they performed, the mention of positive debugging indicators, such as the ability to identify errors or understand the error messages, was related to perceived performance ($\chi^2(1) = 34.17, p < .001$), self-efficacy ($\chi^2(1) = 26.11, p = .001$), and frustration ($\chi^2(1) = 16.81, p = .032$). Students mentioning positive debugging behaviour tended to report higher levels of perceived performance, self-efficacy, and lower levels of frustration. Additionally, the mention of one's self-efficacy within the programming domain or more generally, which was always in a negative fashion, had a significant association with perceived performance ($\chi^2(1) = 18.91, p = .001$) and self-efficacy ($\chi^2(1) = 20.44, p = .009$), reporting lower levels of both.

5 DISCUSSION

The results presented reveal relations between different attitudes and emotions towards debugging. In particular, the prominence of self-efficacy and one's perceived performance is highlighted, with significant correlations to the majority of other consistent constructs. As mentioned, two of the four determinants for self-efficacy originally proposed by Bandura [1] are performance accomplishment and emotional arousal in early experiences of trying a new skill. Therefore, if novice programmers struggle to successfully debug, they will tend to form a negative self-efficacy around debugging. This is more likely to be the case given the emotional reactions often associated with unsuccessful debugging [13, 18, 28]. The link between frustration and anxiety is further established by Lishinski et al. [23] and their strong correlation in the survey responses. Such experiences are not unlikely when students are first learning to debug in a text-based programming language, especially due to the lack of confident computing teachers in many schools [31] who sometime must rush between students [28]. Due to the central role that debugging plays when learning to program, confidence in debugging is likely to be related to confidence in programming generally, meaning that initial struggles with debugging may also affect attitudes aside from those explored in the survey.

The significantly lower levels of self-efficacy in debugging reported by girls corroborates with previous findings on gender disparities in attitudes towards computing [4, 20] and programming [23]. Discussions around improving girls' emotional experiences with programming [10] and sense of belonging in computing [2, 29] are well underway. The pertinence of this disparity in attitudes within the more fine-grained skill of debugging may be the case within other areas of typical computing curricula. If one considers debugging to be an instance of troubleshooting [17, 21], a difference in attitudes towards troubleshooting, a skill extending beyond computing, may also exist.

The range of debugging strategies mentioned by students is not surprising and some have indeed been mentioned by undergraduate novices in previous studies. The inspection of code has been referred to by students in [12], while trial and error has been mentioned by students in [7] and observed by researchers in [38]. Trial and error in particular has been deemed as ineffective and time-consuming [5], making successful debugging less likely. However, no associations were found between this strategy and the attitudes and emotions included in the survey, perhaps due to the low number of students who mentioned it. When comparing other strategies with the survey constructs, the use of external resources was the only one that was negatively related to any of the survey constructs, while mentions of running the code were associated with higher self-efficacy and perceived performance. However, it is important to bear in mind that some strategies were only cited by a few students and it is unlikely that students mentioned every strategy they used.

Other free-text responses were found to be related to some of the emotions and attitudes that students reported. Some students mentioned their lack of self-efficacy in computing more generally, which, unsurprisingly, was linked with their self-efficacy and perceived performance.

5.1 Limitations

The main limitation of this study was that the survey used was not externally validated. Although the decision to create a survey was appropriately justified (see Section 3.2), some parts of it yielded low internal consistency, which impacted the degree of analysis. Additionally, the survey only contained two questions per construct it was aiming to measure. More questions may have improved the overall internal consistency of these constructs.

Despite the results revealing some correlations between students' attitudes and emotions, the number of participants was too low for other exploratory analysis methods, such as factor or cluster analysis, to be successfully employed, instead yielding clusters that were consistent with the correlation matrix. The sample of participants was also skewed in terms of gender, further limiting the conclusions that can be made about the gender-related correlations.

Finally, responses to the free-text questions were typically a sentence in length. It is likely that students in this age bracket are not able to fully nor accurately express their debugging behaviour, meaning they were not all-encompassing responses.

6 CONCLUSIONS AND FUTURE WORK

This paper has investigated the attitudes and emotions that lower secondary students have towards debugging. Results of the exploratory analysis on a survey answered by 73 students indicated that thoughts and feelings towards debugging are, similar to programming, interlinked, making it a somewhat polarising skill to learn. Self-efficacy and perceived performance on the debugging exercises were found to be correlated with multiple other constructs, highlighting the important role of self-efficacy in relation to other attitudes in debugging. Additionally, female students reported lower levels of self-efficacy when debugging, corroborating with findings from related studies. Students also reported a range of debugging strategies, with some initial associations between certain strategies and attitudes and emotions towards debugging found.

This research has further emphasised the attitudinal and emotional struggles of debugging for introductory programmers [13, 18, 28, 36]. Future research must consider how to teach debugging in a way that effectively manages the harmful emotions that school and university students so often experience. Other future work includes the validation of the survey presented in this paper, which is important for measuring attitudes and emotions surrounding debugging in future and more large-scale studies. The next phase of the study will involve analysing the log data generated by students when attempting the debugging exercises, to investigate students' debugging behaviour in more detail.

ACKNOWLEDGMENTS

The authors acknowledge the support of the teachers and students involved in the study, without whom this research would not have been possible.

The first author would also like to give thanks and glory to the Lord Jesus Christ for blessing them, among many other things, with the wisdom and knowledge to carry out this study.

REFERENCES

[1] Albert Bandura. 1978. Self-efficacy: Toward a unifying theory of behavioral change. *Advances in Behaviour Research and Therapy* 1, 4 (1978), 139–161. https://doi.org/10.1016/0146-6402(78)90002-4

[2] Behaviour Insights Team. 2022. *Gender Balance in Computing; Evaluation of the i3 Belonging Intervention.* Technical Report. Behaviour Insights Team. https://www.raspberrypi.org/app/uploads/2023/02/Gender-Balance-in-Computing-Evaluation-Report-Belonging.pdf

[3] Yoav Benjamini and Yosef Hochberg. 1995. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Source: Journal of the Royal Statistical Society. Series B (Methodological)* 57, 1 (1995), 289–300.

[4] Sylvia Beyer. 2014. Why are women underrepresented in Computer Science? Gender differences in stereotypes, self-efficacy, values, and interests and predictors of future CS course-taking and grades. *Computer Science Education* 24, 2-3 (7 2014), 153–192. https://doi.org/10.1080/08993408.2014.963363

[5] Sharon McCoy Carver and David Klahr. 1986. Assessing Children's Logo Debugging Skills with a Formal Model. *Journal of Educational Computing Research* 2, 4 (11 1986), 487–525. https://doi.org/10.2190/KRD4-YNHH-X283-3P5V

[6] Sharon M Carver and Sally C Risinger. 1987. Improving children's debugging skills. In *Empirical studies of programmers: second workshop*. Ablex Corporation, Norwood, New Jersey, 147–171. https://www.researchgate.net/publication/262311647

[7] Mei Wen Chen, Cheng Chih Wu, and Yu Tzu Lin. 2013. Novices' debugging behaviors in VB programming. In *Proceedings - 2013 Learning and Teaching in Computing and Engineering, LaTiCE 2013*. IEEE, Macau, Macao, 25–30. https://doi.org/10.1109/LaTiCE.2013.38

[8] Louis Cohen, Lawrence Manion, and Keith Morrison. 2017. Descriptive statistics. In *Research Methods in Education* (8 ed.). Routledge, New York, Chapter 40, 753–775. https://doi.org/10.4324/9781315456539-40

[9] Louis Cohen, Lawrence Manion, and Keith Morrison. 2017. Statistical significance, effect size and statistical power. In *Research Methods in Education* (8 ed.). Routledge, New York, Chapter 39, 739–752. https://doi.org/10.4324/9781315456539-39

[10] Maggie Dahn and David DeLiema. 2020. Dynamics of emotion, problem solving, and identity: Portraits of three girl coders. *Computer Science Education* 30, 3 (7 2020), 362–389. https://doi.org/10.1080/08993408.2020.1805286

[11] Brian Dorn and Allison Elliott Tew. 2015. Empirical validation and application of the computing attitudes survey. *Computer Science Education* 25, 1 (1 2015), 1–36. https://doi.org/10.1080/08993408.2015.1014142

[12] Sue Fitzgerald, Reneé McCauley, Brian Hanks, Laurie Murphy, Beth Simon, and Carol Zander. 2010. Debugging from the student perspective. *IEEE Transactions on Education* 53, 3 (8 2010), 390–396. https://doi.org/10.1109/TE.2009.2025266

[13] Jamie Gorson, Kathryn Cunningham, and Marcelo Worsley. 2022. Using Electrodermal Activity Measurements to Understand Student Emotions While Programming. In *ICER '22: Proceedings of the 2022 ACM Conference on International Computing Education Research*. Association for Computing Machinery, New York, 105–119. https://doi.org/10.1145/3501385.3543981

[14] Shuchi Grover and Roy Pea. 2013. Computational Thinking in K–12. *Educational Researcher* 42, 1 (1 2013), 38–43. https://doi.org/10.3102/0013189X12463051

[15] Patricia Haden. 2019. Descriptive Statistics. In *The Cambridge Handbook of Computing Education Research*. Cambridge University Press, Cambridge, UK, 102–132. https://doi.org/10.1017/9781108654555.006

[16] Nicole J-M Blackman and John J Koval. 2000. Interval estimation for Cohen's kappa as a measure of agreement. *Statistics in Medicine* 19, 5 (3 2000), 723–741. https://doi.org/10.1002/(SICI)1097-0258(20000315)19:5<723::AID-SIM379>3.0.CO;2-A

[17] Irvin R. Katz and John R. Anderson. 1987. Debugging: An Analysis of Bug-Location Strategies. *Human—Computer Interaction* 3, 4 (1987), 351–399. https://doi.org/10.1207/S15327051HCI0304{_}2

[18] Paivi Kinnunen and Beth Simon. 2010. Experiencing Programming Assignments in CS1: The Emotional Toll. In *Proceedings of the Sixth International Workshop on Computing Education Research*. Association for Computing Machinery, New York, NY, USA, 77–85.

[19] Tobias Kohn. 2019. The Error Behind The Message: Finding the Cause of Error Messages in Python. In *SIGCSE 2019 - Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. Association for Computing Machinery, Inc, New York, NY, USA, 524–530. https://doi.org/10.1145/3287324.3287381

[20] Hayley C. Leonard, Oliver Quinlan, and Sue Sentance. 2021. Female pupils' attitudes to computing in early adolescence. In *Proceedings of the 2021 Conference on United Kingdom I& Ireland Computing Education Research (UKICER '21)*. Association for Computing Machinery, New York, NY, USA, Article 5, 6 pages. https://doi.org/10.1145/3481282.3481289

[21] Chen Li, Emily Chan, Paul Denny, Andrew Luxton-Reilly, and Ewan Tempero. 2019. Towards a Framework for Teaching Debugging. In *ACM International Conference Proceeding Series*. Association for Computing Machinery, New York, 79–86. https://doi.org/10.1145/3286960.3286970

[22] Alex Lishinski and Aman Yadav. 2019. Motivation, Attitudes, and Dispositions. In *The Cambridge Handbook of Computing Education Research*. Cambridge University Press, Cambridge, UK, 801–826.

[23] Alex Lishinski, Aman Yadav, and Richard Enbody. 2017. Students' emotional reactions to programming projects in introduction to programming: Measurement approach and influence on learning outcomes. In *ICER 2017 - Proceedings of the 2017 ACM Conference on International Computing Education Research*. Association for Computing Machinery, Inc, New York, NY, USA, 30–38. https://doi.org/10.1145/3105726.3106187

[24] Gregory R. Maio and Geoffrey. Haddock. 2009. *The Psychology of Attitudes and Attitude Change*. Sage Publications, London. 276 pages.

[25] Lauri Malmi, Judy Sheard, Päivi Kinnunen, Simon, and Jane Sinclair. 2020. Theories and Models of Emotions, Attitudes, and Self-Efficacy in the Context of Programming Education. In *ICER 2020 - Proceedings of the 2020 ACM Conference on International Computing Education Research*. Association for Computing Machinery, New York, NY, USA, 36–47. https://doi.org/10.1145/3372782.3406279

[26] Lauren Margulieux, Tuba Ayer Ketenci, and Adrienne Decker. 2019. Review of measurements used in computing education research and suggestions for increasing standardization. *Computer Science Education* 29, 1 (1 2019), 49–78. https://doi.org/10.1080/08993408.2018.1562145

[27] Philipp Mayring. 2000. Qualitative Content Analysis. *Forum Qualitative Sozialforschung* 1, 2 (6 2000), 1–9. https://www.proquest.com/docview/867646667/1084108BAE714808PQ/2?accountid=9851

[28] Tilman Michaeli and Ralf Romeike. 2019. Current Status and Perspectives of Debugging in the K12 Classroom: A Qualitative Study. In *IEEE Global Engineering Education Conference, EDUCON*. IEEE Computer Society, Washington, DC, 1030–1038. https://doi.org/10.1109/EDUCON.2019.8725282

[29] Allison Mishkin. 2019. Applying self-determination theory towards motivating young women in computer science. In *SIGCSE 2019 - Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. Association for Computing Machinery, Inc, New York, NY, USA, 1025–1031. https://doi.org/10.1145/3287324.3287389

[30] Roy D Pea. 1986. Language-Independent Conceptual "Bugs" in Novice Programming. *The Journal of Educational Computing Research* 2, 1 (1986), 25–36.

[31] Royal Society. 2017. *After the reboot: computing education in UK schools*. Technical Report. The Royal Society. https://royalsociety.org/-/media/policy/projects/computing-education/computing-education-report.pdf

[32] Janet Seeley Blouin. 2011. *High school seniors' computer self-efficacy and interest in computer science careers*. Ph.D. Dissertation. University of Georgia.

[33] Rebecca Smith and Scott Rixner. 2019. The Error Landscape: Characterizing the Mistakes of Novice Programmers. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, Vol. 7. ACM, New York, NY, USA, 538–544. https://doi.org/10.1145/3287324.3287394

[34] Juha Sorva. 2023. Misconceptions and the Beginner Programmer. In *Computer Science Education: Perspectives on Teaching and Learning in School*, Sue Sentance, Erik Barendsen, and Carsten Schulte (Eds.). Bloomsbury Academic, London, UK, Chapter 20, 259–273.

[35] David L. Streiner and Geoffrey R. Norman. 2011. Correction for multiple testing: Is there a resolution? *Chest* 140, 1 (7 2011), 16–18. https://doi.org/10.1378/chest.11-0523

[36] Jacqueline Whalley, Amber Settle, and Andrew Luxton-Reilly. 2021. Novice Reflections on Debugging. In *SIGCSE 2021 - Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. Association for Computing Machinery, Inc, New York, 73–79. https://doi.org/10.1145/3408877.3432374

[37] Aman Yadav, Sarah Gretter, and Susanne Hambrusch. 2015. Challenges of a computer science classroom: Initial perspectives from teachers. In *ACM International Conference Proceeding Series*, Vol. 09-11-November-2015. Association for Computing Machinery, New York, NY, USA, 136–137. https://doi.org/10.1145/2818314.2818322

[38] Wei Yan, Maya Israel, and Tongxi Liu. 2021. Elementary Students' Debugging Behaviors in a Game-based Environment. In *ICER 2021 - Proceedings of the 17th ACM Conference on International Computing Education Research*. Association for Computing Machinery, Inc, New York, 441–442. https://doi.org/10.1145/3446871.3469792

[39] Daniel Yekutieli and Yoav Benjamini. 1999. Resampling-based false discovery rate controlling multiple test procedures for correlated test statistics. *Journal of Statistical Planning and Inference* 82 (1999), 171–196. www.elsevier.com/locate/jspi