Computer programming skills: A cognitive perspective

Graafsma, Irene

DOI:
10.33612/diss.168003240

Publication date:
2021

Summary

Summary

This thesis examines computer programming from a cognitive perspective. Programming is a relatively new skill that has become increasingly important throughout society in recent years. In order to be able to teach this skill optimally, it is essential to understand the cognitive nature of programming. Specifically, it is important to investigate whether this skill relates to other cognitive skills, personality traits and brain processing, and if so how. This thesis covers all three of these areas.

Chapter 1 provides an overview of the history of programming research. It covers the main findings in the three areas of interest and discusses methodologies that are commonly used in these fields. It then states the main research questions of the thesis and introduces the topics covered in of each of the following chapters.

Chapter 2 of the thesis lays the foundations for Chapters 3 and 4 by validating two short versions of the SCS1, a programming test for beginning undergraduate students in computer science. We found that the two short versions could not be considered fully parallel, and that one version was of questionable quality. However, the other version was of comparable reliability and validity to the original, full-length, test, that had been validated in previous studies. Hence, this study results in a validated short version, allowing this test to be used in other research that is only able to allocate short amounts of time to testing.

Chapter 3 explores which of the cognitive skills tested at the start of a semester-long undergraduate programming course, predicted programming performance at the end of that course. We used two measures of programming skill: course-related programming skill, which was measured using the students' course grades, and generalised programming skill, which was measured using the short versions of the programming test described in Chapter 2. We found that logical reasoning skill was the most reliable predictor of programming skill, as it predicted both course-related and generalised programming performance. Algebra and vocabulary learning skills only predicted generalised programming performance. Grammar learning and pattern recognition skills did not predict programming performance on either of the measures.

Chapter 4 investigates whether autistic traits in undergraduate students predict their programming skill at the end of the course. We used the same participants and the

same programming measures as in Chapter 3. We found that the students in our course had higher autistic traits than the general population. However, autistic traits did not predict performance on either of the programming measures. They also did not correlate with any of the cognitive skills described in Chapter 3. This led to the conclusion that autistic traits do not predict programming aptitude. However, it is possible that autistic traits may relate to an interest in programming. This would explain the higher autistic traits in our student population. Future studies will need to investigate this hypothesis further.

In Chapter 5 investigates whether the brain processes syntax violations in a programming language (Java) in a similar way to grammar violations in a native (Dutch) and a foreign (English) natural language. We used Event-Related Potentials (ERPs) as a method to measure electrical brain responses to stimuli. In all three languages, sentences with violations elicited more positive brain responses than sentences without violations. However, there were differences between the languages with regard to the onset, offset and scalp distribution of the effects. Specifically, there was an early onset and offset of the effect in Java, as well a frontal and bilateral scalp distribution. This suggests that this type of bracket violation is processed differently to the subject-verb violations in the two natural languages. Based on both the timing and the scalp distribution, it is inconclusive whether the effect for Java reflects a processing mechanism such as that used for natural languages. However, the effect in response to the bracket violations is similar to that observed in the past in response to orthographic violations in natural language, suggesting that programmers may perceive these violations as incorrect spelling rather than incorrect syntax.

Chapter 6 discusses and integrates the findings of the separate studies and gives suggestions for future studies and for education. Based on the results of the separate studies, logical reasoning seems to be the most reliable predictor of programming performance. Future studies will have to further disentangle the relationships between cognitive skills, teaching and testing methods and programming skills. In particular, they should study whether language skills play a larger role in programming courses where syntax is taught and assessed more explicitly. For education, it may be beneficial to teach logical reasoning explicitly. Additionally, both educators and researchers should be aware

that teaching and testing methods may favour students with certain cognitive strengths over others.

The results of this thesis provide essential new knowledge on the cognitive nature of programming and facilitate future research into this area. Ultimately, the findings in this field will shape our understanding of programming as a skill.