Talking at Cross Purposes - Perceived Learning Barriers by Students and Teachers in Programming Education

Norbert Dorn
Technical University of Munich, Germany
norbert.dorn@tum.de

Dino Capovilla
Humboldt University Berlin, Germany
dino.capovilla@hu-berlin.de

Marc Berges
Technical University of Munich, Germany
berges@tum.de

Peter Hubwieser
Technical University of Munich, Germany
peter.hubwieser@tum.de

ABSTRACT
The knowledge on freshmen’s problems and learning barriers in programming can help the instructors to adapt their teaching and to provide necessary assistance. This paper presents the results of two surveys conducted on secondary school teachers and computer science first-year students even before their first lecture. In total, we interviewed 61 teachers and 61 students with an open-ended questionnaire resulting in a qualitative study of the answers. The goal of the investigation is to find out the perceptions of learning barriers related to programming education. Our results indicate differences as well as commonalities in both perspectives given by teachers and students.

CCS CONCEPTS
• Social and professional topics → Student assessment; • Applied computing → Education;

KEYWORDS
learning barriers, programming education, professional development

ACM Reference Format:

1 INTRODUCTION
As described in many studies, programming is a difficult and complex task [8, 17]. Therefore, many students drop out of their studies of computer science within the first few semesters [9, 10], which might have different reasons, e.g., specific learning barriers in programming.

During the last years, an increasing amount of studies around pedagogical content knowledge (PCK) and content knowledge (CK) in computer science education was published [1, 7, 15, 18], which imply the necessity to take into account students’ cognitive processes during teaching. To enable teachers to cope with this problem during their classes, they have to be aware of such learning barriers. However, how do students perceive the problems they face during learning to program? Moreover, do teachers think about students’ problems in the same way?

To obtain a probable result on these questions, secondary school teachers, and first-year computer science students (before their first lecture) were asked how they perceive learning barriers regarding learning to program. The study is intended to list any learning barriers recognized by both groups as well as those noted by only one of the two groups.

2 RELATED WORK
Milne and Rowe conducted a study on perceived difficulties of concepts related to programming in 2002 [13]. They did an online survey with students and teachers to rate essential concepts of programming on a scale of 1 to 7. The most challenging concept identified by teachers and students was pointers followed by virtual functions, memory allocation, and polymorphism. The authors stick to a conceptual perspective on difficulties in programming.

In 2004, Bruce et al. published a phenomenographic study on how students experience the act of learning to program [3]. With semi-structured interviews, five categories of experiences were gathered each was widening the focus of the previous one. The students experienced programming as a process of having to follow a precise structure (“Following”). In the next category, the students perceive programming as “Coding” whereas the third category focuses on the understanding and integrating of concepts to programming. Other students experienced programming as “Problem Solving”. The last category includes the way programmers think to be part of learning to program.

A study of Joel Michael on perceptions of science teachers and students related to science classes was conducted in [12]. The main issue of the article is that it is essential to know the learning barriers in a subject to enable teachers to enhance active learning for the participating students. In conclusion, the author calls for more creativity and flexibility to overcome perceived barriers of students and teachers.

Shah et al. conducted an empirical search for barriers to learning computer science concepts [16]. In a structured interview, students
had to answer questions on the runtime of given Java programs containing iterations. The target group was first-semester university students as well as grammar school students on a secondary level. The interviews were coded with the help of qualitative text analysis of Mayring [11]. Finally, the authors mapped their findings to common alternative conceptions found in the literature.

In 2015, Funke et al. investigated teachers’ perception of gender differences in programming classes [6]. They conducted a semi-structured interview with teachers resulting in three categories describing differences between boys and girls. For example, the teachers stated that boys have more self-confidence than the girls. Similarly, our study focuses on teachers’ perceptions which could be shown to be effective when looking for learners’ problems.

The study of 2016 conducted by Funke et al. [5] investigated students’ perceptions of computer science. For that reason, biographic data related to computer science and influences on the choice for major in computer science were systematically analyzed. The focus of the study is again on gender differences. These distinctions have to be borne in mind when interpreting perceptions in general as they can cause a gender bias.

3 METHODOLOGY

For our survey, computer-science grammar-school teachers as well as computer science (CS) first-year students of our university were asked to complete a questionnaire to explore the perceived learning barriers regarding learning to program. The students were asked after attending a preliminary course but four weeks before their first lecture at a university. In total 341 CS first-year students were invited via email to fill in an online questionnaire. The preparatory courses were installed to handle heterogeneity in the CS1 lecture. So, no content of CS1 was anticipated by the course [4]. Besides, the questionnaire was presented to 185 teachers during a CS education symposium. In total 61 teachers and 61 students responded.

Both groups were asked almost the same question, only differing in some wording. The teachers were asked to describe their students’ difficulties based on observations: “What do you think are the difficulties for students to learn programming?”. In contrast, the students were asked to reproduce their own experiences but not their problems. They were asked to reflect problems first-year students might have while considering their difficulties: “For which reasons do you think programming is difficult to learn for some students?”. The interviews were conducted independently for both groups in separate online surveys. In general, the respondents were encouraged to exemplify their answers in a few sentences.

Besides, the participants were asked some other questions on their demographic background. While the students were asked on their prospecting course of studies, and their experience in programming, the teachers were presented questions on their teaching subjects, their teaching experience in Computer Science, and the gender distribution of their computer science students.

All answers from both surveys were analyzed using the qualitative content analysis of Mayring [11]. Two researchers - one for each interview group - developed categories from the content. To ensure coding reliability and coder agreement, the researchers coded 20% of the text segments twice and revisited the differences in a re-coding process. The procedure was repeated until almost no differences occurred during re-coding.

4 RESULTS

4.1 Demographic Data

Regarding the subject with a teaching qualification, the investigated population is homogeneous concerning the qualification in CS (95%). The average teaching experience in CS of the investigated population is nine years. Teachers with up to four years of teaching experience are still inexperienced (28%), whereas those with more than twelve years are considered to be very experienced (23%).

The student participants were recruited from a preparatory course for which students from different courses of study were invited to attend. The majority enrolls for a CS major (59%). All others head either for the Games-Engineering programme (28%), a Computer Science in Economics major (10%), or any other courses of study (3%). Furthermore, students were asked about their prior programming experience by explicitly asking them how many lines the broadest program had that they ever had developed before their attendance of the preliminary course. To conclude, the population can be said to be almost (90%) inexperienced (<100 lines of code).

4.2 Perceived Difficulties

The resulting coding system is shown in Table 1. The first two columns present the results from both surveys separately. The last two columns name the subgroups and groups built from the presented categories. The categories were put into three groups (I-III) with Group I separated into three subgroups (Ia-Ic), Group II separated into two subgroups (IIa-IIb), and Group III separated into two subgroups (IIa-IIb). Some of the categories from the teachers’ survey were coded more detailed and mapped to a single category in the students’ survey. In Table 1 this is expressed by the curly brackets. There are some categories written in bold type-case which says that they are most frequently mentioned by the respective group (teachers or students).

The most frequently mentioned difficulties from the teachers (Col. 1) are programming language/syntax and diligence/commitment/stubbornness followed by motivation/interest, organizational factors, modeling/OOM, algorithms/control structures, and abstraction capacity. The other problems perceived by teachers are partly singular, such as cheating, “homework is done on facebook or by dragging and dropping from the neighbor’s USB stick”.

On the other hand, most of the students (Col. 2) perceive the way of thinking as one of the problems novice programmers have. These five categories (abstract thinking, complex thinking, logical thinking, a new/unfamiliar way of thinking, and special thinking) are marked with an asterisk (*) in Table 1. For example, a student’s answer in these categories was that the “thinking in programming is special”. Similarly, the terms “logical”, “abstract”, “complex” and “unfamiliar” were associated with the way of thinking. Other frequent answers were categorized with syntax, missing prior knowledge, and commitment.

For better understanding, the categories are grouped into three groups (Col. 4). The categories in which teachers and students agree

1 All quotations have been translated from German into English by the authors.
Table 1: Comparison and grouping of the two coding systems from the surveys

<table>
<thead>
<tr>
<th>Teachers</th>
<th>Students</th>
<th>Subgroup</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>cheating</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>a gap in basic knowledge</td>
<td>-</td>
<td>-</td>
<td>Ia</td>
</tr>
<tr>
<td>performance differences</td>
<td>-</td>
<td>-</td>
<td>Ia</td>
</tr>
<tr>
<td>structured thinking/overview/problem solving</td>
<td>-</td>
<td>-</td>
<td>Ia</td>
</tr>
<tr>
<td>excessive demand</td>
<td>-</td>
<td>-</td>
<td>Ia</td>
</tr>
<tr>
<td>variables</td>
<td>-</td>
<td>-</td>
<td>I</td>
</tr>
<tr>
<td>context/practice reference</td>
<td>-</td>
<td>-</td>
<td>Ib</td>
</tr>
<tr>
<td>references</td>
<td>-</td>
<td>-</td>
<td>Ib</td>
</tr>
<tr>
<td>algorithms/control structures</td>
<td>-</td>
<td>-</td>
<td>Ib</td>
</tr>
<tr>
<td>modeling/OOM</td>
<td>-</td>
<td>-</td>
<td>Ib</td>
</tr>
<tr>
<td>organizational factors</td>
<td>-</td>
<td>-</td>
<td>Ic</td>
</tr>
<tr>
<td>lack of self-confidence</td>
<td>reservations</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>diligence/commitment/stubbornness</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>keep up with time</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>programming language/temporal effort</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>exactitude</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>concentration</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>diligence/commitment/stubbornness</td>
<td>-</td>
<td>-</td>
<td>Ila</td>
</tr>
<tr>
<td>motivation/interest</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>abstraction capacity</td>
<td>abstract thinking</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>error messages</td>
<td>-</td>
<td>-</td>
<td>Ila</td>
</tr>
<tr>
<td>programming language/syntax</td>
<td>debugging</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>programming language/complexity</td>
<td>syntax</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>complex thinking</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>logical thinking</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>a new/unfamiliar way of thinking</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>special thinking</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>missing prior knowledge</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>communication</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>mathematics</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>divide and conquer</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

The categories written in bold type are the most frequently mentioned by the respective group (teachers or students).

* These categories are primarily based on the way of thinking.

** Due to the answers that were given by the teachers in this category, the categories could not be matched with only of the students’ categories commitment or timely engagement.

in their perceptions are put together in Group II; perceptions mentioned by teachers and not by students are in Group I; perceptions mentioned by students and not by teachers are in Group III.

The subgroups indicated by a contain the categories that relate to general characteristics of the learners. For example, a teacher recognizes a kind of lack of self-confidence (“fears of contact”) and a student mentioning a “fear of the matter and of breaking something” showing reservations. The categories listed in subgroups b relate to conceptual knowledge. For example, a teacher wrote that problems are caused by “understanding and using error messages”, and a student perceives that “the most difficult is to locate the error if you can’t find out where the error is or how to fix it.”. Subgroup c reflects the external conditions. For example, according to a teacher who proclaims that there are “too little time/hours” provided by the curriculum.

5 DISCUSSION

Particular attention is paid to the similarities and differences between the two groups surveyed. The following observations only focus on the categories with a compelling consensus within the groups (bold typeface in Table 1).

First, the problems that are exclusively perceived by teachers (Group I) should be considered. Organizational factors (Group Ic), such as the curriculum, timetables, or the exercises given in the textbooks, can only be perceived or criticized by experts. Additionally, teachers perceive difficulties in Group Ib: “in the younger grades the algorithmic basics are not available, there is more try + error. Even higher grades have problems with this.” (algorithms/control structures), or “class <– object” (modelling/OOM). In contrast, students perceive difficulties in the way they are thinking (Group IIa). So, students’ perceptions focus more on cognitive facets while teachers stick to conceptual issues.

In contrast, from the students’ perspective, the only point from the conceptual knowledge (Subgroup b) that is not singular and is additionally in line with the teachers, is syntax in subgroup Ib. So, a teacher recognizes the difficulty in the fact that “the smallest errors lead to a termination” and a student sees in the syntax a “secret language”. Notably, this category is independently considered by both groups to be one of the most important and, therefore, there seem to be serious learning barriers, which is in line with the literature [13, 16, 17].

The main difficulties the teachers state and that are in line with the students’ point of view are programming language/syntax, diligence/commitment/stubbornness, and motivation/interest. For example, teachers perceive the difficulty in the fact that the students “[have] no patience and perseverance when something doesn’t work right away”, or “students partly have a lack of motivation to program themselves”. From a student’s sight, they “resign [...] facing
According to our results, most of the students perceive the way of thinking that "lack of abstraction". Accordingly, students describe that programming demands the "ability to think abstractly". Again, abstract thinking is seen as one of the most critical aspects of learning to program.

6 CONCLUSION

According to our results, most of the students perceive the way of thinking as the problem that novice programmers have to face. On the other hand, teachers mainly perceive content knowledge and commitment as the fundamental problems of the students.

Notably, the commitment points to a problem John Biggs describes in his levels of teaching competences [2]. The teachers focus on what the student is instead of fostering what the student does, which necessitates further investigation in the professional development of in-service and pre-service computer science teachers. More precisely, the intervention in programming education as a part of pedagogical content knowledge is the focus of our next research.

Especially the categories of group lb can result from a small self-efficacy regarding algorithms and modeling which is following the second level of Biggs. The teachers perceive students’ problems as a lack of own competencies.

The focus of the teachers on content knowledge rises an interesting question: which subject-specific content and which competencies can be assigned to the students’ five ways of thinking (special, logical, abstract, complex and new/unfamiliar). Again, further research is needed to answer this question. In particular, it needs to be clarified whether the different ways of thinking are five different items or whether there is any overlap; perhaps it is even just one single item. Furthermore, the described ways of thinking seem to correspond to different conceptions of computers in general as described in [14].

Besides the differences in the perceptions we have found, there is a consensus in the perceptions of teachers and students. More precisely, all those perceived learning barriers, which are observed by teachers, are in fact also perceived as problems by the learners themselves. So, there are problems where teachers should continue to focus on, integrating the students.

So, in the end, the perceptions of students and teachers have to converge to enable good teaching and learning. For this reason, prospective professional development has to focus on the found differences which have to be deeper elaborated. On the one hand, the teachers have to be enabled to face students problems on a conceptional base regarding the concepts of programming. On the other hand, the way of thinking in computer science has to be put into focus in teachers professional development which is part of student cognition in pedagogical content knowledge related to computer science as described in [7].

REFERENCES