Universiteit Leiden
The Netherlands

# Opleiding Informatica

The Teaching Methods and Strategies of Programming Education within High Schools in The Netherlands

Maxime van Dongen

Supervisors:
Anna van der Meulen & Giulio Barbero

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

12/08/2024

**Abstract**

This thesis will explore the different teaching methods and strategies within the field of programming education in high schools in the Netherlands. The results were gathered through a survey distributed under computer science teachers. With the goal of getting more in-dept information, two computer science teachers participated in online interviews. Programming is such a unique skill to learn, since it consists of theoretical and practical aspects. A mix of instructional methods seems to be implemented by many teachers in programming education. This mix contains direct instruction as well as discovery learning. Different factors determine the required teaching method. Furthermore, most teachers use a combination of pre-existing teaching materials and self-created teaching materials.

# Contents

# 1 Introduction

The shortage of teachers in both primary and secondary education has been a regularly recurring news item in the Netherlands. A news article [Unk21] from November 2021 mentions that the largest shortage in high school teachers in the Netherlands occurs in the field of computer science (at the primary and high school level commonly referred to as programming education). The prognosis for the year 2030 is that there will be an increase to a 40% shortage in computer science teachers. Remarkable is that from 2015 to 2019, the percentage of high schools offering computer science as a subject has decreased, while nowadays the increasingly digital world is an almost inevitable topic.

This thesis is about teaching methods and strategies in programming education, for which there are three areas to connect. Those three areas can be listed as follows: *programming education in high school*, *methods of instruction* and *teachers*. The first area of programming education in high school refers to the school subject of computer science or another school subject that is related to programming. The second area, methods of instruction, is about the strategies and models for education that are used by teachers. Finally, the area of teachers refers to the chosen perspective of this research, meaning that the focus is on the perspective of teachers and not the perspective of students.

The relevance of this thesis is demonstrated by the fact that articles about research in programming education in the Netherlands are limited although research into this subject is conducted more often abroad. Especially the implementation of mandatory core objectives, commissioned by the government regarding digital literacy shows the importance of this topic [SLOc]. Although digital literacy is broader than programming, one of the domains in the curriculum is *Computational thinking*, which strongly relates to programming education [SLOa].

## 1.1 Research question

The main research question for this thesis is: **"Which methods for instruction do teachers use for programming education in high school and how do they experience these methods?"** There are three sub questions that support the main research question, namely:

- "Which teaching methods exist and are used by teachers?"
- "What do teachers think is effective in these methods and what do they know about the experiences from their students?"
- "What could potentially be improved about these methods?"

## 1.2 Thesis overview

This chapter (Section 1) gives an introduction of the subject and states the research question. Section 2 contains the theoretical background regarding teaching strategies and programming education. The used methods for this thesis can be found in Section 3 and the result are described in Section 4. Section 5 contains a discussion of the findings and further research suggestions. Finally, the conclusion can be found in Section 6.

# 2 Background

An ongoing debate exists in the context of teaching programming education between direct instruction and discovery learning. The presumed perspectives in previous research on whether or not these strategies are recommended for teaching programming highly differ. Both direct instruction and discovery learning are general teaching strategies or educational methods that are not specifically designed for programming education. The methodologies in teaching programming are constantly improving and often different methods are applied [DM17]. The reason behind combining methodologies is the fact that learning how to program demands studying the theoretical aspects as well as practising programming as a practical skill. This chapter will discuss different teaching strategies followed by an overview of computer science and programming in high school.

## 2.1 (Explicit) Direct Instruction

The term direct instruction was first introduced by Rosenshine in 1976 [GWD86]. Direct instruction refers to a teaching strategy which has a focus on the behaviour of teachers and their classroom organisation [HS18]. The central principle of direct instruction is described as a teacher-led method where the teacher determines what happens and in what order that happens. The name direct instruction can be explained by the fact that the teacher directs information to students. Lessons that follow the principle of direct instruction are structured, sequenced and the learning goals are clear. The level of strictness that teachers pursue the method of direct instruction differs. Direct instruction does not directly entail any curriculum issues, but there is a particular focus on how instructional time is used [GWD86]. Further, for all students to learn, the key principle in design for direct instruction is that both the materials and presentation of the teachers should be clear and unambiguous.

After the introduction of the term direct instruction many variations on this teaching strategy arose, such as "explicit direct instruction" [HY12]. Hollingsworth and Ybara introduced explicit direct instruction in 2009. In their follow-up book they described explicit direct instruction to be practices to design and deliver powerful lessons that explicitly teach content to all students from all grade-levels. Another core aspect of this instruction model is to continuously check the understanding of the material with the students. Explicit direct instruction is a blueprint where the goal for the students is to obtain more knowledge during instruction in the classroom.

To specify the meaning of direct instruction in programming education, the activities of teachers when using this method in programming courses are summed up as follows [DM17]: "Lecturing, presenting and reviewing programme elements and algorithms; Instructions on learning programme syntax; Demonstration on completed solved programmes." Here, the activities of the students are defined as: "Interactive lectures following; Asking questions regarding elements and algorithms and completed solved programmes." From this, it can be deducted that there is a difference in the role of teachers and students in terms of programming education following direct instruction. Where the teacher has to provide clear instructions and learning goals, the students are expected to pay attention to the lectures and be critical in asking their questions. To elaborate, the example of "completed solved programmes" comes back in the role of the teacher as well as the role of the student. Teachers are required to demonstrate and explain them and students should be asking questions and actively participate in interactive lectures.

## 2.2 Discovery Learning

A different approach to instruction in the classroom is discovery learning and can also be referred to as constructivism or exploratory learning [HS18]. This has been a dominant strategy for science and mathematics [Coh08]. In opposite of direct instruction, students take a major part of the responsibility for their learning process themselves in discovery learning. The role of the teachers is minimal in discovery learning with the belief to allow students to gather their knowledge independently [HS18]. Typical activities in the classroom regarding discovery learning include role playing, group projects and computer simulations [Coh08].

Research shows that computer scientist are often unaware of their didactic beliefs, since they are not used to discussing those [HS18]. According to this article, discovery learning is the one didactic philosophy that many programmers do believe in and there are three reasons to explain this believe. The first reason is based on Piaget who believes that children build their knowledge through assimilation and accommodation. Secondly, there is no collective memory of how programming was taught, since most programmers taught themselves. Many teachers are programmers who repeat their own experiences resulting in exploratory lessons. The third reason is the belief that programming is not for everyone and people “with programming genes” can teach themselves by exploring.

## 2.3 Direct Instruction versus Discovery Learning

For more than over half a century, there has been a debate on what the ideal amount of instruction and guidance should be in the classroom [Coh08]. In this article it is stated that researchers Bruner (1961), Papert (1980) and Steffe & Gale (1995) support the idea of discovery learning. The same article states that researchers Cronbach & Snow (1997), Klahr & Nigam (2004), Mayer (2004) and Sweller (2003) advocate direct instruction as a strategy over discovery learning.

Although discovery learning is widely used, many articles present resistance against this approach. Previous research supporting direct instruction over minimal guidance describes that based on the current view on human cognitive architecture, it is unlikely for minimally guided instruction to be an effective teaching method [KSC06]. The main reason being the change in view of long term memory, where this is now seen as the central and dominant structure of human cognition. It is explained that nothing is learned when nothing is changed in the long term memory. Long-term memory is seen as a “massive knowledge base that is central to all of our cognitively based activities”. Working memory is where conscious processing takes place. When new information is being processed, the working memory is limited in both capacity and duration. Instructional methods depending on minimal guidance are said to place a huge burden on working memory. This load on the working memory is unable to make a contribution to accumulating knowledge in the long-term memory. Long-term memory is unavailable as long as the working memory is used for searching solutions to problems. Another article supporting direct instruction ([CKS12]) concludes that teaching methods with minimal guidance only benefit the most expert students. They support explicit instruction over students attempting to discover knowledge themselves. It is also stated here that studies almost uniformly support instructions to be full and explicit. Especially when looking into the novice and intermediate students, explicit instruction is said to be favoured over partial and minimal guidance for them. Students who already have prior knowledge are in advantage in exploratory teaching, since they are better at asking the right questions and elaborate their existing knowledge [HS18]. The students appreciating exploratory learning the most, namely the weaker students are said to especially suffer in this teaching strategy when looking into performances of students.

Previous research in support of discovery learning describes that there are three interrelated conditions to make discovery learning an effective method [Coh08]. The first condition entails the meaning of discovery learning. Students need activated prior knowledge to be able to understand the problem and to help them create hypotheses. Second on the list is logicality of the activities, since discovery learning includes proper scientific reasoning and the ability to manipulate variables. Students should learn rules and principles from situations that can be applied to other settings, since reflective generalisation is the last condition from this list. Previous research that compares direct instruction with discovery learning referred to the findings of Klahr & Nigam (2004), and mentioned that Klahr & Nigam (2004) do not consider the maintenance of knowledge over time [DJK07]. As an argument against direct instruction it is stated that direct instruction is unnecessary and insufficient when a longer period of time is taken into account.

## 2.4 Gamification

One final teaching method to introduce in this section is gamification. Gamification is a relatively new teaching concept. Whereas direct instruction and discovery learning are strategies for instruction, gamification can be seen as an additional tool for teachers to use in their lessons. The concept of gamification is to use a game (element) in the curriculum to contribute to the motivation of the students.

A study from December 2021 recommends the integration of gamification in programming education [GR21]. For this research, a gamified course was set up for teaching programming in Python. The students in the experimental group scored significantly higher than the students in the control group. The experimental group also obtained higher scores in terms of knowledge, but this was not significantly. The students in the gamification course had significantly higher scores for attitude and self-efficacy compared to the students in a traditional setup.

## 2.5 Computer Science and Programming in High School

A research into the perspective of teachers in computer science gives an overview of the challenges and experiences from teachers in the United States [YGHS16]. The goal of understanding these challenges comes from the increasing demand in teachers, the risk of teachers dropping out and the lack of clearness in programs to obtain a teaching license. It is mentioned here that beginning computer science teachers struggle to meet pedagogical demands caused by lack of knowledge in both computer science content and pedagogical knowledge. The term PCK (*Pedagogical Content Knowledge)* was first introduced by Shulman in 1986 and connects the areas of content knowledge and pedagogical knowledge, since they highly depend on each other. It is mentioned that studies into computer science PCK are sparse and little is known in this area. The article sums up three themes in which the beginning teachers struggle, namely: content challenges, pedagogical challenges and assessment challenges. Content challenges refer to the limited knowledge of teachers regarding computer science and the relationship with programming experience leading beginning teacher to challenges in exploring concepts in depth which affects the learning of their students. Pedagogical challenges contain the student-centered nature of computer science which makes it difficult to keep all students focused. Another pedagogical challenge is the uniqueness of problem solving in computer science, which makes it difficult to address all students' needs. The final pedagogical challenge is the complexity of meeting the needs of students on an individual basis. Lastly, the assessment challenges refer to the lack of assessment tools, the collaborative nature of assignments which makes it if difficult to accurately evaluate knowledge of individual students, and using rubrics to grade which can reduce creativity and expression. Other mentioned themes in challenges faced by beginning computer science teachers include lack of teacher preparation, solitude in working as a computer science teacher, and IT challenges, for example, unavailability of the latest technologies.

Another debate in programming education is on which programming languages should be used. Research from 2021 into which programming languages should be used to teach programming describes that a generation of programming languages is dynamic and evolutionary [Sob21]. This means that the most popular programming language varies per year and everyone has their own preferences in languages. It is concluded here that there is no consensus on the question which programming language is the best to introduce programming to students and it is mentioned that there probably never will be consensus.

Specifically concerning the situation in the Netherlands, computer science became a school subject for the first time in 1998 in Dutch high schools [Grg21]. Currently, there is no national exam for computer science, nor has this ever been the case. From the beginning until 2019, there did not exist a national curriculum for this course. The curriculum still gives teachers a lot of freedom which they strongly appreciate. This does not contain a mandatory programming language or advise on which programming language should be used, but only demands students to be able to program in an imperative programming language [SLOb].

## 2.6 Definitions

To conclude this section, some definitions are needed that apply to the remainder of this thesis. When *students* are mentioned, this refers to high school students attending the Dutch education system on the level of *Havo* or *VWO*. Usually their ages range from 12 to 18 years old. The term *programming education* includes both the school subject of computer science as well as other school courses that are related to programming.

# 3 Methods

This research used a survey about teaching strategies in programming education. Participants of this survey were teachers who teach programming and/or computer science in high school in the Netherlands. The goal of this survey was to map and get a picture of teaching strategies in programming education in Dutch high school education. Besides distributing the survey for this research and to expand the knowledge and data for this research, there were two interviews with computer science teachers. This section will describe the process of the data collection, starting with the procedure in Section 3.1, followed by describing the participants in Section 3.2 and then the measurements will be discussed in Section 3.3, this contains a description of the contents of both the survey and the interviews. To conclude this Section, the analysis of the data will be discussed in Section 3.4.

## 3.1 Procedure

This research was reviewed by the ethical committee. The survey was created in Qualtrics and filling out the survey was anonymous. The language of the survey was Dutch and the interviews were in Dutch. A post on an online forum for teachers of computer science and/or programming education in the Netherlands, called i&i [ien], asked teachers to fill out the survey. Besides that, the board of the i&i forum included a request to fill in the survey in their newsletter. Informaticavo [inf] also posted the survey in its newsletter. Informaticavo is, similar to the i&i forum, an organisation for computer science teachers in the Netherlands. Finally, acquaintances who are working as teachers in high schools in the Netherlands received an email that asked for their participation in the survey. The messages on these forums and the emails included the request to sent out the survey to other computer science teachers in their network. The survey was open from the 11th of April 2023 to the 31st of July 2023 and in total there were 24 respondents.

Both of the interviews were online through Zoom. Before starting the interview, the participants received a form to sign for permission, the so called informed consent, to use the interview in this research. The first interview was on the 19th of June 2023 and the second interview on the 11th of July 2023.

## 3.2 Participants

A total of 24 respondents to the survey and the volunteers for the interviews and their backgrounds will be discussed. Starting with the participants for the survey, their ages ranged from 22 to 69 years old. Both the median and the average age of these participants came to 47 years of age. From the 24 participants, 18 of them were male, 4 of them were female and 2 of them identified as other. Asking about the years of experience, there were 3 options to answer this question. The first option was 0-5 years experience, the second option 5-10 years experience and the final option was more than 10 years experience. Within the range of 0-5 years experience, there were 7 of the participants, in the range of 5-10 years there were 2 participants and the other 15 participants had more than 10 years of experience. There were 11 respondents who also teach other courses in high school or also teach higher education, where the other participants said to not teach other courses or left this answer open. Figure 1 provides a summarised graphical overview of the background of the participants. More information about the background and other courses taught by the participants, can be found in section 4.1.1.

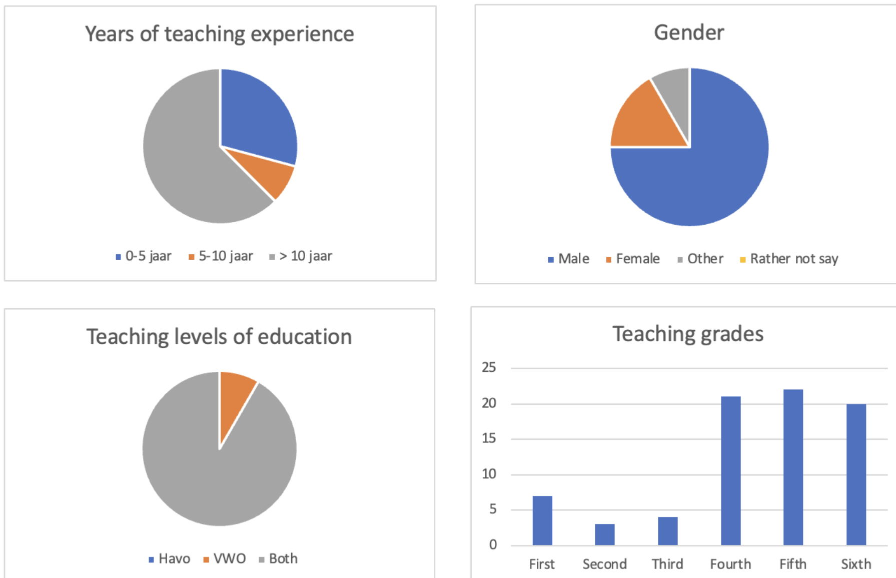


Figure 1: Background of particpants survey

Continuing with the participants of the interviews, both of the participants were experienced in teaching computer science in high school. Participant A also taught in higher education, specialising in Embedded Systems. Participant B was a teacher trainer besides teaching high school students. Additionally, the two participants were both experienced in the research field.

## 3.3 Measurements

The survey consisted of five sections and the complete survey can be found in Section A. In all of the sections, except for the introduction, there was a non-mandatory field where participants could leave a comment or explain their given answers. Some responses to the questions are already discussed in Section 3.2, but for completeness, this section contains a brief overview of the content of the survey. In the first section, the introduction of the survey, the questions asked about age, gender, experience and background. Further, there were questions asking if the participants also taught different courses and which levels of high school (*Havo*, *Vwo* or both) and which years (first to sixth) they currently teach. Section 2 was about teaching materials and methods. The questions about teaching materials asked participants about programming languages and materials used in their lessons. Regarding the teaching methods, it was asked what a typical start of a lesson entails. Section 3 contained statements relating to teaching strategies and instructional methods. Participants were asked to respond using a 5-point Likert scale ranging from ‘never’ to ‘always’. These statements were translated and adapted from an article about programming education and teaching quality by Jia and Hermans (2022) [JH22]. Section 4 was designed to ask participants about their level of contentment regarding programming education and teaching materials. Participants were asked to rate this on a 7-point scale, where 1 refers to 'very discontent' and 7 refers to 'very content'. Section 5 was about the level of effectiveness regarding teaching methods and materials. There were five statements where the participant had to answer on a 5-point scale about the level of contentment referring to methods, instruction and knowledge of the students.

The article used to formulate the statements in Section 3 of the survey ([JH22]), describes their research for which a survey was distributed to teachers in programming education in China. The situation in programming education in China as described in the article is comparable to the situation in the Netherlands. The majority of the statements from Jia and Hermans were direct translations to Dutch for use in this survey. In total, there were 10 statements in this section of the survey. After both statement 5 and statement 10, there again were open, non-mandatory questions for remarks and elaboration if needed. All statements can be found in Figure 2.

The goal of the interviews was to elaborate on the questions in the survey and obtain more in-dept knowledge on these questions. The interviews were semi-structured and the preparation of these interviews resulted in a list of follow-up questions next to the questions in the survey. This list mainly contained questions asking to elaborate their answers and to illustrate their experiences. There was enough room for the participants to give their opinions and add other topics. The first interview had a duration of approximately 50 minutes and the second interview had a duration of approximately 20 minutes.

## 3.4 Analysis

A direct export of the data from Qualtrics to Microsoft Excel made it possible to use Microsoft Excel for analyzing the data and creating the graphs. The preproccesing of the data will be discussed in Section 4. For measuring the internal consistency in the sections from the survey about teaching methods and strategies, but also about effectiveness, the Cronbach's alpha needed to be calculated and can be found in Table 1.

| Direct Instruction | |
|---|---|
| **Item** | **Cronbach's alpha** |
| **All** | **0.401** |
| **Delete 1** | 0.273 |
| **Delete 2** | 0.392 |
| **Delete 3** | 0.374 |
| **Delete 4** | 0.369 |
| **Delete 5** | 0.316 |
| **Delete 6** | 0.398 |
| **Delete 7** | 0.453 |
| **Delete 8** | 0.416 |
| **Delete 9** | 0.336 |
| **Delete 10** | 0.395 |

| Effectiveness | |
|---|---|
| **Item** | **Cronbach's alpha** |
| **All** | **0.674** |
| **Delete 1** | 0.532 |
| **Delete 2** | 0.586 |
| **Delete 3** | 0.554 |
| **Delete 4** | 0.731 |
| **Delete 5** | 0.680 |

Table 1: Chronbach's alpha for Direct Instruction and Effectiveness

Since calculating the Cronbach's alpha was not possible for the results of the section regarding contentment, this section needed the Pearson correlation coefficient ($r$). The value of $r$ came to $-0.260$ which corresponds with a weak and negative correlation.

To analyse the interviews, the transcribed interviews needed to be marked to summarise the necessary information. A selection of these quotes needed to be classified per topic and per participant. Combining these quotes in a new file resulted in a table that contained an overview of the quotes of both of the participants that were classified per topic.

# 4 Results

Within this section the results per paragraph of the survey will be discussed, but first the preproccesing of the data will be described. The first step in processing the results is the preprocessing of the data. First, the preprocessing of the data from the survey will be discussed, followed by discussing the process for the interviews. Preprocessing the data from the survey started with deleting the incomplete responses, since all of these participants ended too early in the survey to use their information in this research. Subsequently, the data from Qualtrics was directly exported to Microsoft Excel. Per step in the preprocessing of the data, a separate file in Microsoft Excel was created to maintain an overview of the conducted steps. For the questions where participants were asked to answer on a scale and there was a possibility to leave a comment, separate sheets for the closed and open answers were created. The reason for this was to ease creating the graphs representing the data and to maintain a clear view of the results. In order to process the results from the interviews, for both of the interviews the recordings were transcribed in separate files. After that, both of the interviews were summarised. Finally, a file was created that contained a table where selected quotes from the participants were classified per topic.

The results per section of the survey will be discussed in Section 4.1. The final section of this chapter, Section 4.2 contains the results from the interviews.

## 4.1 Results from the Survey

As per section of the survey, the results will be presented in the next subsections. Each section of the survey can be found in a separate section, namely Section 4.1.1 contains the section of the introduction of the survey, Section 4.1.2 contains the section of the methods of the survey, Section 4.1.3 contains the section direct instruction of the survey, Section 4.1.4 contains the section about contentment of the survey and finally, Section 4.1.5 contains the section about effectiveness of the survey.

### 4.1.1 Introduction

As discussed in Section 3.2, the participants who filled in the survey ranged from ages 22 to 69, with both the median and the average age being 47 years of age. Table 2 shows the responses from the first section of the survey. This table contains the answers to the questions regarding gender, years of teaching experience, teaching levels of education, teaching grades and teaching other courses on the side.

From the participants who also teach other courses, two of them teach physics and two other participants teach mathematics in high school. One teacher also teaches geography in high school. Another teacher teaches the subject *Onderzoek & Ontwerpen*, which literally translates to research and design. The last respondent who also teaches another course in high school teaches biology, information technology (vmbo) and creative techniques. The remainder of respondents who also teach other courses teach in higher education in the following courses: Industrial Design, Computer Science, Embedded Systems, and lastly Technical Drawing and Autocad.

| Variables | Category | Frequency | Percentage |
|---|---|---|---|
| **Gender** | Male | 18 | 75.00% |
| | Female | 4 | 16.67% |
| | Other | 2 | 8.33% |
| | Rather not say | 0 | 0.00% |
| **Years of teaching experience** | 0-5 years | 7 | 29.17% |
| | 5-10 years | 2 | 8.33% |
| | More than 10 years | 15 | 62.50% |
| **Teaching levels of education** | *Havo* | 0 | 0.00% |
| | *VWO* | 2 | 8.33% |
| | Both | 22 | 91.67% |
| **Teaching grades** | First | 7 | 29.17% |
| | Second | 3 | 12.50% |
| | Third | 4 | 16.67% |
| | Fourth | 21 | 87.50% |
| | Fifth | 22 | 91.67% |
| | Sixth | 20 | 83.33% |
| **Teaching other courses on the side** | Yes | 11 | 45.83% |
| | No or unanswered | 13 | 54.17% |

Table 2: Background of respondents (N=24)

#### 4.1.2 Methods

In this section about methods, the questions asked about used programming languages and methods in the lessons of the participants. The results from the closed questions from this section, which asked about teaching methods and the typical start of a lesson, can be found in Table 3.

| Variables | Category | Frequency | Percentage |
|---|---|---|---|
| **Teaching methods** | Pre-existing teaching method or textbook | 4 | 16.67% |
| | Created my own teaching material | 7 | 27.17% |
| | Combination of pre-existing material and created material by myself | 13 | 54.17% |
| **Typical start of a lesson** | Students get an instruction at the start | 16 | 66.67% |
| | Students start with working on their own | 8 | 33.33% |

Table 3: Teaching methods and typical start of a lesson (N=24)

Noticeable is that in all cases where teachers answered to use a pre-existing method as teaching material, the lessons were usually started with an instruction. From the 13 teachers who use a combination, 5 of them usually start their lessons with the students working on their own, where the other 7 of them usually start with an instruction. From the 7 teachers who exclusively use teaching material created by themselves, there are 3 who have their students working on their own at the beginning of a lesson and the other 4 start their lessons with an instruction.

The one open question in this section asks which programming languages are used. All of them mentioned Python, except for one of the participants. That one participant did mention the use of a pre-existing method which has Python as its default programming langugage, so with that information it can be deducted that all of the participants do use Python in their lessons. Several other languages were also mentioned here, namely (not in particular order): CSS, HTML, PHP, Swift, Hedy, Javascript, Scratch, Micro:bit, C++, MakeBlock, LUA, p5js, C#, Godot, Haskell, NetLogo, Processing and MySQL.

### 4.1.3 Teaching Strategies and Instructional Methods

Figure 2 shows both the statements and given answers from this section of the survey. On the left hand side of the figure, the statements are presented. The coloured blocks on the right hand side represent the number of times an answer was given to that specific statement. For example, looking at the third statement it follows from the graph that none of the participants said they never or rarely explain to their students what they expect them to learn, and that by far most of them do this often or always.

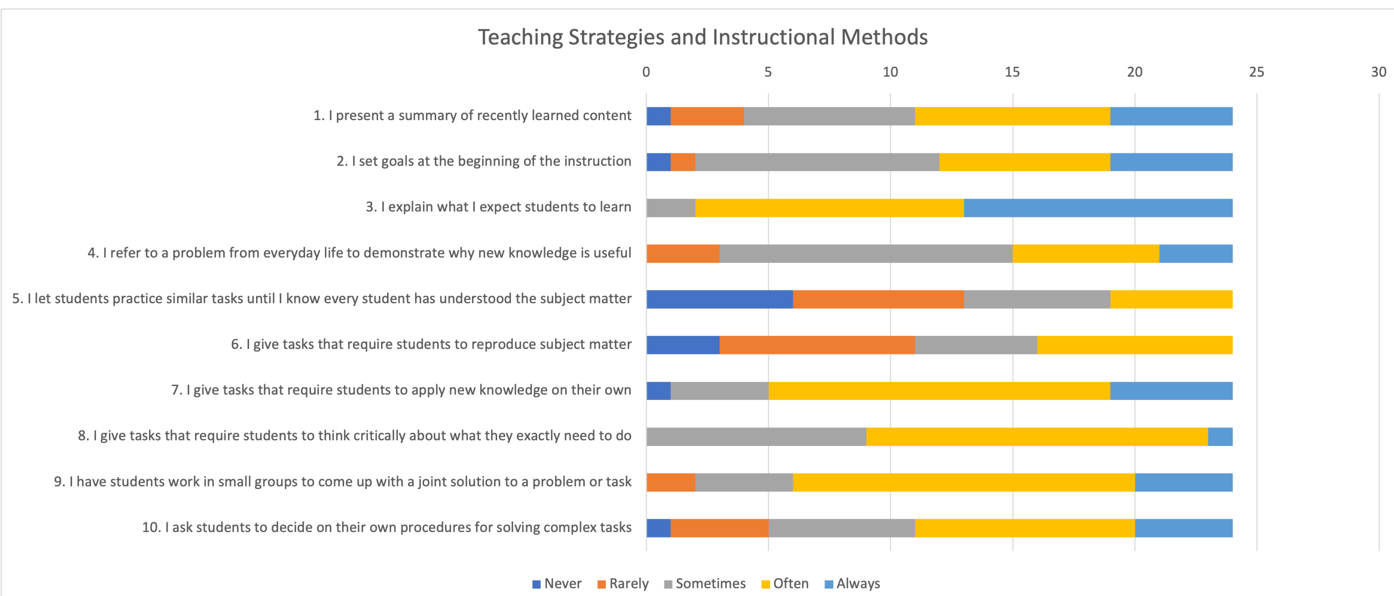


Figure 2: Teaching Strategies and Instructional Methods

The subject of statement 1 up to and including statement 4 regard the instruction part of the teaching strategy of direct instruction. In these four statements the participants are asked whether they implement characteristic features in their instructions. These features are presenting a summary of recently learned content, setting goals, explaining the expectations and demonstrating why the subject matter is meaningful. A recurring remark was that in some schools the students pick their own topic to work on and decide themselves what their learning goals are.

Statement 5 is about the teaching pace and asks whether the participants let their students practice similar tasks until the moment they are sure all of the students understand the subject matter. As can be seen in the graph, the answers to statement 5 differ the most. This was also one of the statements to which some respondents added a remark. A remark given multiple times in this section is that the level of students differ and thus the teachers lower the expectations for some students or teachers do not focus on the slowest students when deciding the pace. It is also said here that students sometimes work on their own pace.

Statements 6, 7 and 8 are about the tasks the participants give their students. Statement 6 asks whether participants give tasks that require students to reproduce the subject matter. Remarkable is that none of the participants answered ‘always’ to this statement, and that the majority of the participants answered ‘never’, ‘rarely’ or ‘sometimes’. Statement 7 asks if the participants give their students tasks that require application of knowledge on their own. In statement 8, the participants are asked whether they give tasks that require their students to think critically about what they exactly need to do. The majority of around 60%, that is 14 out of 24, of the participants answered ‘often’ to both statement 7 and statement 8.

Statement 9 asks if the participants let students work in small groups in order to have them come up with a joint solution to problems or tasks. Noticeable here is that none of the participants answered ‘never’ and again the majority of around 60% answered with ‘often’ to this statement.

Concluding with statement 10 about letting students decide on their own procedures for solving complex tasks, the answers are more scattered. One of the participants commented that this statement was unclear. Another participant added that students often do not know how to decide on their own procedures. Is was also stated here that that the more skilled students usually decide on their own procedures. To complete the given remarks in this section of the survey, one of the participants stated that the way of teaching does not really fit this questionnaire.

#### 4.1.4 Contentment

This section contained two closed questions regarding contentment. The first question asked about the level of contentment regarding programming education and the second question about the level of contentment regarding teaching material. Both questions had answering options ranging from 1 to 7, with 1 representing ‘very uncontented’ and 7 representing ‘very content’. Table 4 shows the results from the question regarding contentment in programming educations as well as contentment in teaching material.

| Variables | Category | Frequency | Percentage |
|---|---|---|---|
| **Contentment programming education** | 1 | 0 | 0.00% |
| | 2 | 2 | 8.33% |
| | 3 | 0 | 0.00% |
| | 4 | 3 | 12.50% |
| | 5 | 8 | 33.33% |
| | 6 | 8 | 33.33% |
| | 7 | 3 | 12.50% |
| **Contentment teaching material** | 1 | 1 | 4.17% |
| | 2 | 1 | 4.17% |
| | 3 | 2 | 8.33% |
| | 4 | 2 | 8.33% |
| | 5 | 7 | 29.17% |
| | 6 | 8 | 33.33% |
| | 7 | 3 | 12.50% |

Table 4: Contentment regarding programming education and teaching material

It is noteworthy that there were two participants who rewarded their contentment regarding programming education with a 2 and rewarded their contentment regarding teaching material with a 6. Other noteworthy combinations of answers are that the level of contentment regarding programming education was given a 7 three times, while they gave the contentment regarding materials once a 3, once a 5 and once a 6.

Participants differ strongly in opinion in this section of the survey. Where one teacher wrote that all levels of knowledge in advance do not make a difference and all students can get by, another teacher wrote that the number of students from classes 1 to 3 who are less and less skilled with the result that the desired level of knowledge is achieved less often. One participant wrote that the focus is too much on self discovery instead of good instructions.

#### 4.1.5 Effectiveness

The statements and given answers from this section are shown in Figure 4.1.5. The question here asks the participants to what extent they agree with five statements regarding effectiveness of their teaching methods. The answering scale ranges from 1 to 5, where 1 represent 'strongly disagree' and 5 represents 'strongly agree'. Noticeable about these answers is that 'strongly disagree' was never given as an answer, as can be seen in Figure 4.1.5, and the answer 'disagree' was only given once in statement number 4. The Cronbach's alpha score for this section is significantly higher than the score for the statements in Section 4.1.3. As illustrated by the yellow blocks in Figure 4.1.5, for most of the statements the majority of the participants answered with 'agree'. The only statement where this is not the case is statement number 4.

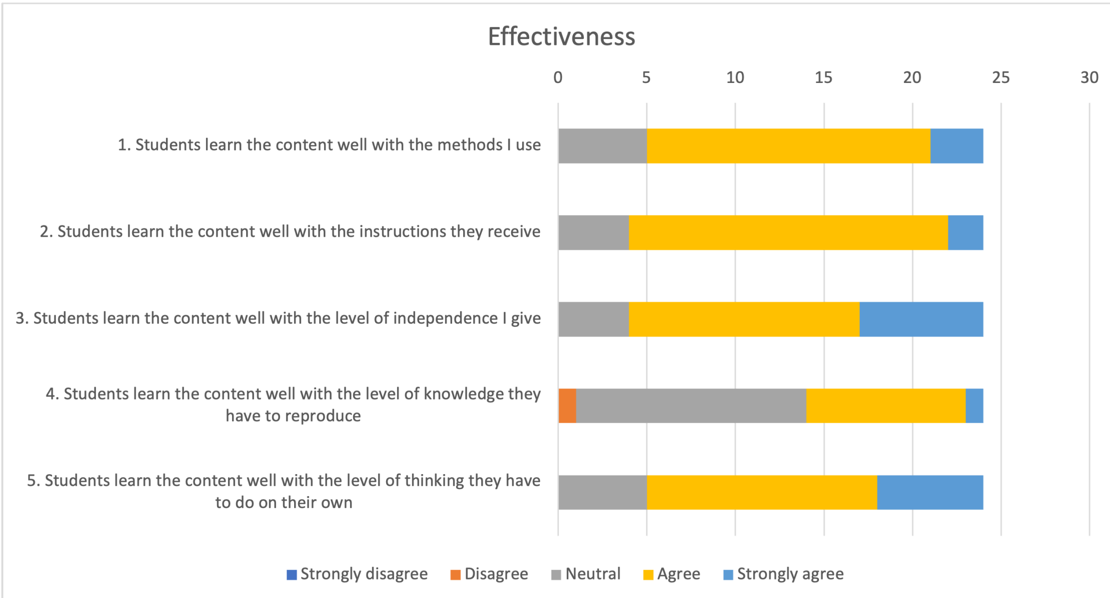


Figure 3: Effectiveness

Comments for this section illustrate that in the opinion of the participants the effectiveness also depends on the commitment of the students and that some students are better in managing independence than others. One participant said that learning how to think as a programmer is a time consuming process and the calmness for this process is not always available. One other participant added in the comments that every student needs to discover what works for them. Another participant stated that there is a good balance between instruction and applying the knowledge.

## 4.2 Results from the Interviews

In Section 3.2, the participants of the interviews were shortly introduced. Here, the results from the interviews will be discussed. The questions from the survey were used as a guideline for these interviews. In the following subsections, the answers will be discussed per subject for both of the participants.

### 4.2.1 Background Participants

The two participants both have experience in teaching high school students in the subject of computer science. However, currently participant A mostly teaches students in higher vocational education (*hoger beroepsonderwijs in Dutch*) at a university of applied sciences, whereas participant B is now teaching computer science at a high school and besides that also works as a teacher trainer. She had worked corporate jobs before obtaining her educational master's degree with a teaching qualification which allows teaching computer science to all years of high school. She first introduced computer science as a course at the high school she is currently working, since they did not have a teacher available then.

Another similarity in the two participants is that they are both experienced in the research field. Besides doing a PhD research, participant A teaches short modules at high schools nearby the university of applied sciences. The topic of the modules is robotics and also includes a social aspect, industrial and medical robotics and ethics. He referred to the modules as a Trojan Horse, since the students get a tour at the university of applied sciences and are shown some projects from students in higher education. Participant B mentioned that she continued teaching computer science while working on her PhD research in subject didactics in computer science.

### 4.2.2 Teaching Material and Tools

Participant A as well as participant B makes use of pre-existing teaching methods in their lessons besides self-written teaching material. Participant A adds to the conversation that in co-operation with a colleague of his, he used to contribute in writing content of licensed pre-existing methods. Participant B says to also gather material from the internet. This includes material from both Dutch and international teachers in computer science. She adds that for the final year of high school, she does not use the licensed pre-existing method, but uses the material composed by the SLO.

Participant B states that she uses Python as a base in her lessons, there are however other programming languages she also uses, namely HTML/CSS - which she does not consider a programming language - and NetLogo when she teaches about computational science. She indicates that other languages are possible when students are interested in learning them.

When it comes to teaching tools, Participant A prefers to work with a teaching environment where the options for tests and exams are integrated for both summative and formative testing. Participant A feels that tools like H5P-layering a video with questions to answer for the students before they are able to continue watching the video are valuable in teaching. He believes that just watching a video does not teach that much to the students. What he also appreciates at tools similar to H5P are the learning analytics which allow the teacher to gather more insight into the learning progress of their students.

### 4.2.3 Teaching Strategies and Instructional Methods

Participant A and participant B agree that there are different factors to determine the required way of teaching and establishing a lesson. However, the factors they consider are different; participant A refers to the characteristics of the students and also mentions a difference in whether lessons are online or in person, whereas participant B addresses the content of the subject to be taught.

The three factors considered by participant A are discussed in this paragraph and subsequently the next paragraph will discuss the factors that participant B considers. First, when it comes to the factor whether lessons are online or in person participant A states that there is not really any other option than direct instruction. He says to prefer lessons in person. With the online lessons, there are several practical problems, e.g. the students do not always have a soldering iron on themselves where those are always available when teaching in person. Second, concerning the characteristics from students, he believes that mixed teaching methods are required in computer science, since students have many different levels of knowledge and qualities. Some students already have programming experience, which he thinks is characteristic for computer science and is for example not seen in courses as chemistry, but can also be found in creative courses as music and drawing arts. About those experienced students he says that you do not want to slow down those students and other students might consider those students intimidating, especially when they are showing off their skills. Lastly, he addresses that he notices a significant difference in boys and girls when it comes to the need of structure, clearness and clarity in expectations as it can be found challenging to work with an “open assignment”. About this he said: “Some students are really looking for structure [...] You can really see a difference in boys and girls at that point”. He also mentions that students withdraw when there is no clarity. Referring to the needs of girls, he said: “They want to know much better what is expected from them etcetera.” In terms of direct instruction, Participant A is convinced that is it crucial to work with rubrics containing the criteria students are assessed for in assignments.

In this paragraph, the factors considered by participant B will be discussed. She emphasises the need for a different approach in her lessons depending on the subjects to be covered. To clarify, when teaching students to program, she sketches a picture where the students do a lot of work on their own and she just walks around the classroom to answer questions and now and then gives an instruction at the beginning of a lesson. Teaching subjects like security require a different approach according to her. She then for example lets the students debate on topics such as cyber crime or ethics. She mentions that she enjoys these differences in teaching methods and that the methods ask for different ways of preparing the lessons. Coming up with topics to debate on is different than preparing for a programming lesson where she has to await which questions arise from the students during the lesson. To elaborate her point of view more in dept, she sketches a lesson she is planning to teach to the fifth year *VWO* right after the summer holidays. The topic on the agenda is data structures, where it is required to learn how to work with arrays. She will start her lesson with: “Summer holiday is over, you have forgotten everything. You should still somewhat remember how to work with Python, and you all know the game rock, paper, scissors." She says not to explain that much, but to just let the students work on this simple assignment and aims to let her students think on their own on how to approach this assignment. Around halfway through the lessons, she will either draw a flow chart or pseudo code on the blackboard and explain that this is about what the students should have coded at that point. During the lesson when the students are working on the assignment, she walks around the classroom and checks what her students are doing and how she could support individual students. She summarises a typical programming lessons as follows: "You give an assignment and let the students work on their own a lot." Again, she emphasises the difference in approach depending on the topics to be taught: "When a topic demands to be a conversation in the classroom, you switch between explaining the topic and let the students discuss the topic in smaller groups." She concludes that programming lessons are more suitable to differentiate between students than theoretical lessons, since more students really immerse themselves into the material. Programming lessons allow her to see for example which students actually need her help from the beginning and which students are already capable in multiple programming languages and need more challenge.

### 4.2.4 Contentment Material and Methods

Regarding his contentment on teaching material, participant A says that this varies. Speaking about pre-existing methods, he feels that at a certain moment, the level of one of the pre-existing methods was really good, but says that the reason for that is that it was written by actual computer science teachers. About other pre-existing methods he says that some content was too theoretical and could be improved where the important concepts could be highlighted more. He adds that the underlying concepts behind the material in the pre-existing methods are not always explained or not clearly enough explained. When that is the case, he feels that the more advanced students can find their ways, but that this is not necessarily the case for the students who are struggling with programming. Participant A believes that when students do not feel comfortable with programming, they will make different choices. Similar to participant A, participant B says to never be content and is always looking for ways to adjust her teaching methods. She illustrates her contentment regarding teaching material with learning environments where students are only allowed to continue working on their assignments when the previous assignment is completed and working. She points out that as a teacher you are ensured that your students actually complete the assignments, but that students sometimes have no idea what they are doing and have just copied what is already in the assignment. Because of this, she feels that the results from the students can be disappointing. Although learning results can be disappointing, she appreciates the environments where students are obligated to complete the previous assignments before being able to continue the next assignments.

Besides discussing their contentment about teaching materials, the participants were also asked their opinions about contentment in teaching methods. As stated in the previous paragraph, participant B said to always keep adjusting and says to enjoy having two roles, since she can immediately try out new ideas. Both of the participants agree that there are many things to improve when it comes to programming education. Participant A answers that he appreciates that nowadays there are bachelor and master students who are obtaining teaching qualifications, but there is still a lot of work left to do in this area. The answer of participant B is focused on improving the possibility of differentiating and wishes to have more teaching material available for this purpose. She adds that teachers are supposed to differentiate themselves, where you try to come up with assignments for some students who need to scaffold into the materials and learn step by step. On the other hand, she wants more challenging assignments for some other students.

### 4.2.5 Additional Notes and Comments

Both of the participants added comments about subjects besides the questions from the interview that are worth mentioning. Participant A addressed the importance of teaching computer science in high school and participant B wondered about the approach of other computer science teachers. About the importance of computer science in high school participant A said “it is important for students to at least understand what computer science is” and added that they should not have to continue their education in computer science in high school, but teaching it in high school results in a more positive view of computer science according to him. He also pointed out that the duration of the lessons, typically 40 to 50 minutes, is a short amount of time to support and help individual students. Participant B was curious about the approach of other computer science teachers, meaning that she believes many teachers simply hand out teaching material and tell students to “just get going and ask your questions”.

# 5 Discussion

The goal of this study was to find an answer to the main research question: "Which methods for instruction do teachers use for programming education in high school and how do they experience these methods?". This research question was supported by three sub-questions, namely:

- "Which teaching methods exist and are used by teachers?"
- "What do teachers think is effective in these methods and what do they know about the experiences from their students?"
- "What could potentially be improved about these methods?"

In this section, discussing and answering these questions will be done. Section 5.1 and Section 5.2 will discuss the research questions supported by the results from the survey and the interviews. Finally, in Section 5.3 the limitations of this study will be discussed.

## 5.1 Teaching methods

The first sub-question on teaching methods focused both on teaching materials and on teaching and instruction strategies. First, concerning teaching methods, the majority of teachers use teaching material that they have created themselves. This can be concluded from both the survey and the interviews. Most, but not all, teachers not only use self-created materials but combine this with existing materials. In previous research, a survey that asked computer science teachers in the Netherlands about the curriculum of computer science showed that for *Havo* there were 29 teachers using pre-existing material, 14 teachers using their own material and 4 teachers use something else. The results for *VWO* showed 65 teachers using pre-existing material, 41 teachers using their own material and 9 teachers use something else [GTdV23]. There are two important differences in the aforementioned survey from previous research and the survey used in this thesis. First, the survey used for this thesis asked about teaching methods including all grade levels in Dutch high school for computer science and/or programming education. The other survey from previous research specifically asked about electives for only the higher grades. Schools are obligated to make a selection (2 for *Havo* and 4 for *VWO*) of provided topics by the SLO [SLOb]. The second difference is in the formulation of the question. The survey from this thesis had the following options to choose from when asked about teaching methods: "Pre-exiting teaching method or textbook", "Created my own teaching material" and "Combination of pre-existing material and created material by myself". The other survey from previous research had the following options: "SLO", "Textbook" and "Other". Where the results from the other survey from previous research indicate a preference for pre-existing material over self-written material, the results in this thesis showed that the preferred method is a combination of pre-existing teaching material and self-written material. A possible explanation for the different results could be the fact that the survey from previous research did not have the option of a combination of pre-existing and self-created teaching material. It is also noteworthy that for most of the electives there is teaching material available by the SLO that is freely accessible. One of the interviewed teachers explained that for the student from the final year she uses the material from the SLO and not the licensed pre-existing textbook.

The used programming languages for teaching can also be taken in account when speaking about materials. Although the national curriculum for computer science in The Netherlands does not demand a specific programming language, all teachers in this research use Python in their lessons. Only one of the teachers from the survey did not explicitly state to use Python, but this teacher mentioned a pre-existing teaching method that uses Python as default programming language. Both of the interviewed teachers also use Python. Many teachers facilitate other programming languages besides Python, such as CSS, HTML, PHP, Swift, Hedy, Javascript, Scratch, Micro:bit, C++, MakeBlock, LUA, p5js, C#, Godot,Haskell, NetLogo, Processing and MySQL. Looking into literature about used programming languages, there is no overview on which languages are actually used. To illustrate this, the following was stated in a study that gave an overview of computer science in Dutch high schools [GT08]: "One can get an impression of what is going on in classrooms all over the country by taking a look at the quite lively online community on *www.informaticavo.nl*. The growing diversity of topics found there is remarkable. A quick look at the collection of tests on programming submitted to the site, for example, shows that the subject is apparently being taught using Visual Basic, Logo, NQC for Lego Mindstorms, Java, Gamemaker and Delphi." Presumable because of the freedom within the Dutch curriculum, it might be that there is no overview on which programming languages are actually used. Another factor that might make it challenging to maintain an overview of used programming languages is that teachers also create their own materials. The question on which programming language is best to introduce programming to students has not yet reached consensus and it is expected that that probably never will be the case [Sob21].

Second, the teaching strategies and instructional methods will be discussed. The survey gave insights in both the method of direct instruction as well as discovery learning. Both methods, or even a mix from these methods seem to be used. The debate on whether direct instruction or discovery learning is the most effective method for teaching programming has not been settled yet. Previous research shows both arguments in favour and against direct instruction as well as discovery learning [Coh08]. Direct instruction is defined as a teacher-led method where the teacher determines what happens in what order. The principle of direct instruction is that lessons are structured, sequenced and have clear learning goals [HS18]. The answers to the following statement (statement 3) from the survey - "I explain what I expect students to learn" - where almost all teachers answered with "Often" or "Always" can be interpreted as direct instruction being a method used by almost all teachers from the survey. Further, the vast majority of teachers often or always explain students what they expect them to learn. This fits the description of direct instruction. The survey shows that most teachers typically start their lessons with an instruction, which is a characteristic of direct instruction. However, when a teacher requires students to obtain their knowledge through discovery learning, it is also possible for the lessons to start with an instruction. The nature of the instruction simply differs in the two teaching methods. This can be seen as an integration of direct instruction within the strategy of discovery learning. The main idea behind discovery learning is students obtaining knowledge by themselves [Coh08]. A recurring comment in the survey was that in some schools, the students are allowed to select their own topic to work on and they have to decide themselves what their learning goals are. This way of designing the curriculum, where students are allowed to choose their own topics to work on, automatically requires the students to obtain their knowledge themselves. This can be seen as discovery learning, since students gathering knowledge on your own is the key feature of discovery learning.

From the interviews, it became clear that both teachers use direct instruction and discovery learning in their lessons. While factors determining the ways of teaching and designing lessons did not come up in the survey, it was comprehensively discussed in the interviews. Those factors entail characteristics of students and the content of the subjects to be taught. Whether lessons are online or in person can also be seen as a factor to adapt lesson plans to. To elaborate, one of the teachers stated that when a lesson is online, there is no other option than direct instruction. Continuing with the characteristics of students, this can refer to students already having programming experience, but is was also pointed out that there is difference in boys and girls. He believes that girls want to know more what is exactly expected from them. Knowing exactly what is expected in a course is according to direct instruction. The other teacher focused more on the content of a lessons. For example, she sketched an example where she gave her students time to figure out how to program something, but later also discussed this within the classroom. This can be seen as a combination of direct instruction and discovery learning. She explained that the more theoretical subjects are more suitable for direct instruction. It can be derived that the more practical a lesson, the more use of discovery learning compared to direct instruction is applied.

The survey used in this research had the purpose of gathering insights in teaching and instruction methods used in programming education in high schools in the Netherlands. This was not always very unambiguous, since not all of the statements and questions were directly linked to one instruction method. For example, looking at the statement from Figure 2, statements 1 - 3 are clearly linked to the definition of direct instruction. Other statements do not necessarily fit the description of direct instruction, but can be seen as teaching practices. For example, looking at statement 5 - "I let students practice similar tasks until I know every student has understood the subject matter" - gives information about differentiation between students in the classroom.

Overall, based on the answers from the survey where almost all teachers answered to start their lessons with an instruction and also explaining to their students what they expect, it can be concluded that direct instruction is used by most teachers. From the comments in the survey stating that students can sometimes decide themselves what topics to work on, it can be derived that they also use discovery learning. From the interviews, where both teachers explained more in-depth about the methods used in their lessons, it can be concluded that they both use direct instruction as well as discovery learning. A combination of instructional methods is in line with previous research [DM17]. The reason given for a combination of teaching methods is the fact that learning how to program demands both studying the theoretical aspects as well as practising programming skills.

## 5.2 Effectiveness and contentment

Regarding the second sub-question, teachers differ strongly in their opinions on the topic of contentment in programming education and teaching material. To elaborate, where one teacher mentioned that all levels of knowledge in advance make no difference and all of the students can keep up the pace of the lessons, another teacher is convinced that students from classes 1 to 3 are getting less and less skilled when starting with programming education and as a result of that the desired level of knowledge is being achieved less often than before. Comparing the results from the survey when asked about contentment regarding programming education and teaching materials in programming education it showed that overall, the teachers were more critical and slightly less content in terms of programming education compared to teaching methods. A possible explanation for this could be that when asked how content teachers are about programming education, the topics and factors that can be addressed here are broader than when asked about contentment regarding materials. Factors to consider here include knowledge level of students in advance, the option for students to work in their preferred programming language and also the uniqueness of programming or computer science as a school subject. Another possible explanation here could be that since most teachers make use of self-written materials, it is easier to adjust that material to personal preferences. When discussing tools where students are only allowed to continue when they have completed the current assignment, the criticism was that often students just "copy and paste" and learning results can be disappointing.

Regarding the third sub-question, several possible improvements for teaching programming to students came up in this research. As mentioned in the previous paragraph, while tools where students are obligated to complete assignments in order to continue are appreciated, there is a risk where students just copy answers. There was no solution mentioned for this risk. Improving the opportunity for differentiating between students also came up. Most suggestions to improve this came down to having more teaching material available and assignments where students are able to choose their programming language to complete the assignment in.

A report with advises to the government, dated 2014, support the arguments of having more materials available for differentiation [TKG14]. The freedom within the curriculum is also comprehensively discussed in this report. The freedom is seen as a benefit, but at the same time also a pitfall. The pitfall here regards the quality of education. Something that was also discussed in the interviews, and came back as comments from the survey is the education of teachers is the education of teachers in computer science. In terms of teaching material, the report states that for most subjects in high school, teaching material can be in hands of publishers. It is questioned if that is suitable for computer science, and stated that this course demands material in the form of modules. Modules are said to be suitable for differentiating in the course. The connection between high school computer science and follow-up education is also mentioned in this report. This does not come back in this research but can be an interesting subject for further research. All of these topics relate to effectiveness and contentment in programming education.

## 5.3 Limitations

This research had a few limitations. First, it would have been helpful to take the background and education of the teachers into account in this research. This was discussed in the interviews, but not exactly in the survey. The survey contained only one open question about the background of the teachers, but there was no specific question about the education of the teachers. This would have been interesting since there are different routes to a teaching license in the Netherlands.

Remarkable, but not exactly a limitation is that from the participants in the survey, 75% of them was male. The exact number of the ratio male-female teachers in computer science in the Netherlands can not be found, but it might not be surprising that females are underrepresented in this area.

Finally, having a sample size of 24 participants in the survey, the generalisation in this case is limited. However, the interviews provided a more in-dept point of view in all topics discussed in the survey, and also gained insights in topics that were not discussed within the survey.

# 6 Conclusions and Further Research

Programming is such a unique skill since it is possible to obtain this skill all through self study, but on the other hand, there are many scientific underlying principles to learn that support learning how to program. The uniqueness of programming as a course and the scarcity of fully relevant literature make it a challenging subject. The main research question for this thesis was: "Which methods for instruction do teachers use for programming education in high school and how do they experience these methods?" Overall, the teachers that participated in this research prefer to use a combination of pre-existing teaching material and materials created by themselves. Almost all teachers implement the teaching method of direct instruction in their lessons. Besides that, many teachers implement (elements of) discovery learning. The combination of instructional methods could be explained by different factors determining the method of instruction. Those factors include the subject of a lesson and the characteristics of students. A recurring suggestion for improvements in programming education was to have more materials available for differentiation. Some teachers indicated to adapt materials themselves when they are not content with it. From that, it can be deducted that it might be useful to make improvements in the pre-existing teaching methods.

This thesis focused on the perspective of students. For further research, it might be interesting to also take other perspectives into account, for example the perspectives of students, management of schools or publishers for teaching materials. Another consideration for further research could be focusing more on the background of teachers.

# References


[CKS12] Richard F. Clark, Paul A. Kirschner, and John Sweller. Putting Students on the Path to Learning: The Case for Fully Guided Instruction. *The American Educator*, 36(1):5–11, 1 2012.

[Coh08] Marisa T. Cohen. The Effect of Direct Instruction versus Discovery Learning on the Understanding of Science Lessons by Second Grade Students. *NERA Conference Proceedings 2008*, 1 2008.

[DJK07] David Dean Jr and Deanna Kuhn. Direct instruction vs. discovery: The long view. *Science Education*, 91(3):384–397, 2007.

[DM17] Slobodanka Djenic and Jelena Mitic. Teaching strategies and methods in modern environments for learning of programming. 2017.

[GR21] Manuel Garcia and Teodoro F. Revano. Assessing the Role of Python Programming Gamified Course on Students' Knowledge, Skills Performance, Attitude, and Self-Efficacy. 11 2021.

[Grg21] N Grgurina. Getting the picture: Modeling and simulation in secondary computer science education. 2021.

[GT08] Natasa Grgurina and Jos Tolboom. The first decade of informatics in dutch high schools. *Informatics in Education*, 7(1):55–74, 2008.

[GTdV23] Nataša Grgurina, Jos Tolboom, and Bart Penning de Vries. Evaluating the new secondary informatics curriculum in the netherlands: The teachers' perspective. In *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*, pages 155–166. Springer Nature Switzerland Cham, 2023.

[GWD86] Russell Gersten, John R. Woodward, and Craig Darch. Direct Instruction: A Research-Based Approach to Curriculum Design and Teaching. *Exceptional Children*, 53(1):17–31, 9 1986.

[HS18] Felienne Hermans and Marileen Smit. Explicit direct instruction in programming education. In *PPIG*, 2018.

[HY12] John R. Hollingsworth and Silvia E. Ybarra. *Explicit Direct Instruction for English Learners*. Corwin Press, 12 2012.

[ien] Website of the i&i association for computer science teachers. https://www.ieni.org.

[inf] Website of the association Informaticavo for computer science teachers. https://www.informaticavo.nl.

[JH22] Xiaohua Jia and Felienne Hermans. Teaching quality in programming education: the effect of teachers' background characteristics and self-efficacy. In *Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 1*, pages 223–236, 2022.

[KSC06] Paul A Kirschner, John Sweller, and Richard E Clark. Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational psychologist*, 41(2):75–86, 2006.

[SLOa] Website of SLO about digital literacy. https://www.slo.nl/sectoren/vmbo/digitale-geletterdheid-vmbo/digitale-geletterdheid-vo/digitale-geletterdheid/, note = "[Online; accessed 2-May-2023]".

[SLOb] Website regarding the curriculum of computer science in High School for The Netherlands Havo and VWO. https://www.slo.nl/handreikingen/havo-vwo/handreiking-se-info-hv/examenprogramma/. [Online; accessed 5-April-2024].

[SLOc] Website regarding the development of the digital literacy learning objectives. https://actualisatiekerndoelen.nl/digitalegeletterdheid. [Online; accessed 2-May-2023].

[Sob21] Sónia Rolland Sobral. The old question: Which programming language should we choose to teach to program? In *Advances in Digital Science: ICADS 2021*, pages 351–364. Springer, 2021.

[TKG14] Jos Tolboom, Jenneke Krüger, and Natasa Grgurina. Informatica in de bovenbouw havo/vwo: Naar aantrekkelijk en actueel onderwijs in informatica. 2014.

[Unk21] Unknown. Gastdocent moet tekort informaticaleraren opvangen: 'sta hier met klotsende oksels'. *RTL Nieuws*, 11 November 2021. Available at: https://www.rtlnieuws.nl/nieuws/nederland/artikel/5266288/docent-informatica-bedrijfsleven-co-teach (Accessed: April 25th, 2023).

[YGHS16] Aman Yadav, Sarah Gretter, Susanne Hambrusch, and Phil Sands. Expanding computer science education in schools: understanding teacher experiences and challenges. *Computer science education*, 26(4):235–254, 2016.

# A Content of the survey

This appendix contains the content of the survey. The original survey was in Dutch. If applicable, the answering options are listed under the questions. At the end of the survey, there was a non-mandatory field for notes and comments.

## A.1 Section 1: Introduction

- What is your age?
- What is your gender?
  - Male
  - Female
  - Other
  - Rather not say
- What is your background? (For example, "Before teaching, I have worked in a different sector for 10 years")
- For how long have you been teaching?
  - 0 - 5 years
  - 5 - 10 years
  - More than 10 years
- Do you teach other courses besides computer science/programming education? If so, which other courses do you teach?
- Which levels of high school do you teach?
  - *Havo*
  - *VWO*
  - Both
- Which grades do you teach?
  - First year
  - Second year
  - Third year
  - Fourth year
  - Fifth year
  - Sixth year

## A.2 Section 2: Methods

- Which programming languages do you use in your lessons?
- Which methods do you use in your lessons?
  - Pre-existing teaching method or textbook
  - Created my own teaching material
  - Combination of pre-existing material and created material by myself
- Do students typically start a lesson with working on their own or do you typically start with an instruction?
  - Students start with working on their own
  - Students get an instruction at the start
- *Non-mandatory field for comments and explanations*

## A.3 Section 3: Teaching Strategies and Instructional Methods

For the following statements, the answering options were: 'never', 'rarely', 'sometimes', 'often' and 'always'.

- I present a summary of recently learned content
- I set goals at the beginning of the instruction
- I explain what I expect students to learn
- I refer to a problem from everyday life to demonstrate why new knowledge is useful
- I let students practice similar tasks until I know every student had understood the subject matter
- *Non-mandatory field for comments and explanations*
- I give tasks that require students to reproduce subject matter
- I give tasks that require students to apply new knowledge on their own
- I give tasks that require students to think critically about what they exactly need to do
- I have students work in small groups to come up with a joint solution to a problem or task
- I ask students to decide on their own procedures for solving complex tasks
- *Non-mandatory field for comments and explanations*

## A.4 Section 4: Contentment

For the following statements, the answering options were a scale of 1 to 7, where 1 represented ‘very discontent’ and 7 represented ‘very content’.

- What is your level on contentment regarding programming education?
- *Non-mandatory field for comments and explanations*
- What is your level of contentment regarding the used teaching materials
- *Non-mandatory field for comments and explanations*

## A.5 Section 5: Effectiveness

For the following statements, the answering options were: ‘strongly disagree’, ‘disagree’, neutral, ‘agree’ and ‘stronly agree’.

- Students learn the content well with the methods I use
- Students learn the content well with the instructions they receive
- Students learn the content well with the level of independence I give
- Students learn the content well with the level of knowledge they have to reproduce
- Students learn the content well with the level of thinking they have to do on their own
- *Non-mandatory field for comments and explanations*