UNIVERSITY OF TWENTE.

ELAN - Instituut voor Lerarenopleiding en Professionele Docentontwikkeling

Identifying and Adressing Common Programming Misconceptions with Variables—Part II

ir. Rifca M. Peters

Master Thesis

December 11, 2018

In collaboration with:
dr. Danny Plass-Oudebos
Under supervision of:
Nico van Diepen

Commity:
dr. Ingrid Breymann
Wim Nijhuis

Track: Computer Science
Master Educatie en Communicatie in de Bètawetenschappen
(formerly Science Education and Communication)
ELAN, Faculty of Behavioural Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

Imperative programming is considered an important, fun, but also difficult topic of computer science education. It requires learners to develop new ways of thinking and learn new concepts. Problems arise when a concept is not understood well, while progress relays on it. For example, consider variables—one of the basic building blocks of programming—, when this concept is not understood it becomes almost impossible to grasp data manipulation. In preceding work (Plass, 2015) we identified common misconceptions about variables and reported their origin as well as a test to assess existing misconceptions for individual students.

In this report we present and evaluate a interactive video instruction designed to address identified misconceptions. We developed an active, goal-oriented intervention based on a constructivist approach; gradually constructing correct understanding of variables in imperative programming. In a paper-cut stop-motion animation a few lines of code are traces, a voice-over explained what happens while the changes in values are visualized. We evaluated the video with students enrolled in an introductory programming course in secondary education. Misconceptions about variables held by students were assessed before and after watching the video. Afterwards students made less errors, indicating that correct understanding of variables was improved. A major decline was visible for the misconception, originating from mathematics, that statements such as `y = x + 20` denote an equation to be solved. Instead, students showed improved understanding of the meaning of the $=$ symbol and the structure of an assignment statement.

Although further research with a different population and a control group is needed, the current results provide strong indications that the interactive video successfully addressed specific misconceptions about variables held by students.

Contents

Chapter 1

Introduction

Imperative programming is a mandatory subject in the computer science curriculum at secondary education in the Netherlands (Schmidt, 2007; Tolboom, Kruger, & Grgurina, 2014). Moreover, programming can be a tool to develop *21st-century skills* (McComas, 2014) such as problem solving, collaborating, and media literacy (Thijs, Fisser, & van der Hoeven, 2014). However, programming is also a difficult skill to learn because it requires a new way of thinking, being able to generalize and abstract (van Diepen, 2014).

A division may arise between learners that *do* and *do not* 'get it'. This is reflected in student grades that follow a bimodal distribution where most students score either below or above the expected average grade (Figure 1.1). Some may believe that students who score below the average have limited programming capabilities, leading to student drop-out (Robins, Rountree, & Rountree, 2003) or even teachers advising a student to do so. Dehnadi and Bornat (Dehnadi & Bornat, 2006) reported this as the "camel hump" and advocated the existence of a simple programming aptitude test dividing programmers from non-programmers. However, this work was retracted because evidence was lacking for the predictive value for performance (Bornat, 2014; Ferguson, 2014). An alternative cause of the bimodal can be sought in the *learned edge momentum* (LEM). The LEM effect states that if subsequent topics in a course are dependent on previous topics, students who grasp the first topic are more likely to grasp the second, and those who do not grasp the first topic are less likely to grasp the second, and therefore less likely to grasp the third, and so on (Robins, 2010). This highlights the importance of good basic understanding to avoid increasing knowledge gaps between students over time. Nevertheless, the variation in students' expertise levels make it difficult to design course materials and processes that are challenging and interesting for all students (Lahtinen, Ala-Mutka, & Järvinen, 2005). Moreover, teachers and course creators must be aware of the issues that hinder learning progress before they can create materials to overcome them (Herman, Kaczmarczyk, Loui, & Zilles, 2008).

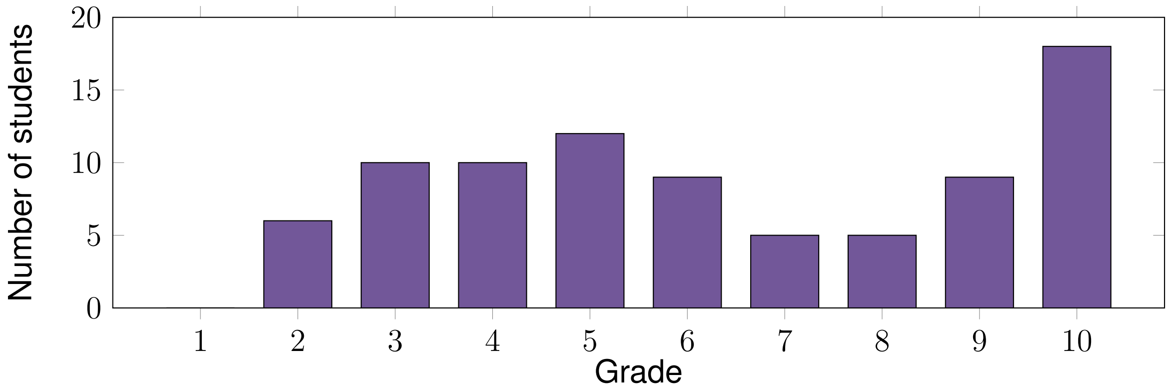


Figure 1.1: bi-modal grade distribution for an introductory course Java programming (HAVO 4, Ludger College, Doetinchem, the Netherlands, 2010-2014)

A short reflecting upon currently available programming lesson materials for secondary education discovers that these provide many exercises, but provide little instruction. This encourages “trial-and-error” practises rather than deep understanding. Moreover, even when provided with comparable, working examples students are not capable of doing the exercises, let alone understand the written code. Students' struggles with programming have been observed in their course work, such as the BMI-calculator assignment (Appendix A). This assignment was designed to assess understanding of different programming constructs. The majority of students displayed poor understanding of the basic construct of variables. For example, students did not convert data correctly to the appropriate type, were unaware of the value of variables at specific moment, and did not use variables whenever opportune (see Appendix A, Section A.3).

In collaborative work, misconceptions about variables in imperative programming amongst younger, novice programmers have been further investigated. In earlier work Plass (2015) presented identified misconceptions and our tests developed to assess misconceptions held by students. In the present work, I describe the material designed to instruct correct understanding about variables. Further, I report on the empirical study done to evaluate the effectiveness of this material.

The remainder of this report is organised as follows. In Chapter 2, we present the identified misconceptions and go into some details of programming didactics. Then, in Chapter 3 we describe the design of the interactive instruction video. Followed by the study methodology in Chapter 4, and the results in Chapter 5. In Chapter 6 we discuss the effectiveness of the video based on the results. Finally, in Chapter 7, conclusions and recommendations are given.

Chapter 2

Background & Related Work

The aim of the present work is to develop an intervention —in the form of an interactive video— teaching correct understanding of variables to novice programmers. In this section we report existing knowledge on three important aspects: current state of programming education, misconceptions about variables, and video instructions. This information serves as the foundation for the design of our instruction material.

2.1 Computer Science Education

Programming is a mandatory subject of the computer science curriculum at secondary education in the Netherlands[1] However, programming is considered a hard subject nonetheless due to the abstract concepts (van Diepen, 2014; Kuittinen & Sajaniemi, 2004). Analogies, such as the container or "box", used to explain these concepts may lead to misconceptions. For example, moving instead of copying a value, and the ability to contain multiple items (Smith, DiSessa, & Roschelle, 1993).

2.1.1 Computer Science Teaching Methods

In the Netherlands are three dominating, published computer science teaching methods (Tolboom et al., 2014), Enigma, Fundament Informatica (Instruct) and Informatica-Actief. Although programming is mandatory, Stichting Enigma Online (2013) is the only method offering a full introductory (Java) programming course in their main curriculum. Instruct (2018) included "concept functions" in their main curriculum and offers supplementary programming modules. INFORMATICA-Actief (2015) included algorithms in their main course. Alternatively, teachers develop their own programming courses (e.g., Programming in Delphi (Heijmeriks, 2007)).

[1] Subdomein B3: Software 7. De kandidaat beheerst eenvoudige datatypen, programmastructuren en programmeertechnieken.

These methods have in common that they tend to focus on procedural rather than conceptual knowledge. Students have to 'write' a full application following stepwise instructions[2] or examples, without having mental models of how this functions. For example, Enigma's Java and the Delphi course instruct OO-programming using a WYSIWYG-editor. Meaning that students create application windows using a visual editor, and then write a few lines of code to add functionality to a button. In this process, variables are used without proper instruction of their function and behaviour. The methods provide examples of the correct syntax for various functionality (e.g., retrieve user input, do some calculation, and write the result on screen) and exercises to apply the new bits and pieces. With exception of Informatica-Actief's algorithm module, which does focus on programming concepts rather than writing syntax. The method uses a visualiser to show the effect of changes applied to variables, loops and subtasks outside a language specific environment.

2.1.2 Constructivism in Computer Science Education

Constructivism is a theory of learning claiming that students actively construct knowledge rather than passively receive and store knowledge presented by a teacher or book (Ben-Ari, 1998). This approach is based on the view of Piaget and Vygotskys (1987), stating that humans construct meaning in the interaction between their experiences and their ideas. Related is Vygotsky's (1980) theory on the zone of proximal development (ZPD), which marks the difference between what a learner can do without help and what they cannot do. It is believed that experiences in the ZPD encourage and advance learning. Meaning that the learning process should be tailored to the learner's prior knowledge and experiences. Key elements are that knowledge builds upon existing knowledge and that one should focus on understanding of essentials rather then learning by heart. A constructivist approach requires advanced instruction skills; a teacher should provide adaptive guidance based on the student's understanding.

Ben-Ari (1998) states that for students with no prior model, the teacher must ensure that a viable hierarchy of models is constructed; meaning this must be explicitly instructed and discussed. Instruction should not be limited to procedural knowledge (to do x, follow steps 1 to n), and exercises should be delayed until there is a viable model constructed. Premature attempts likely lead to endless "trial-and-error" programming, which does not facilitate development of expert-like programming skills. Further, one should be aware that autodidactic prior experiences not necessarily correlate with success, they may as well cause firm non-viable models (i.e., misconceptions).

[2] For an example see the BMI-assignment in Appendix **??**

The constructivist approach showed to support adoption of deep programming strategies and structures, and is recommended for teaching variables (Kuittinen & Sajaniemi, 2004). Adhering to the constructivist approach, Kuittinen and Sajaniemi (2004) recommend to first introduce constants (named literals), then fixed values (constants set at runtime), and one by one introduce dynamic functions such as the stepper (counting) and transformer (calculation). Each of these different roles of variables should be instructed by a description and concrete examples expressing the variable purpose and behaviour. Animations can support explaining various roles by visualising the past and future values, and show the syntax to access or transform values stored in variables. An active role of the student can further improve effectiveness of the animation (Mayer, 1988). Somewhat surprisingly, as described above, most existing teaching methods in the Netherlands don't follow these recommendations.

Although a constructivist approach is preferable, learning outcomes highly depend on the teachers expertise, skills and commitment. Otherwise, students receiving inadequate guidance and support, risk becoming frustrated and discouraged ultimately leading to disengagement and non-adherence (Wilson, 2012). This stresses the importance of high quality, easy to use and well formed (i.e., conforming to the constructive approach) instruction materials to support teachers in their knowledge transfer and student guidance.

2.2 Misconceptions about Variables

Students may hold certain misconceptions about variables. Although misconceptions are —according to constructivism— necessary to construct new knowledge (Smith et al., 1993), they need to be identified and transformed to correct conceptual models in order to facilitate development of programming skills.

2.2.1 Identified Misconceptions

Studies on misconceptions in programming —and about variables in particular— revealed four categories of origin of misconceptions: mathematics, anthropomorphism, analogy, and semantics.

People learn everyday and build upon previous obtained knowledge. However, sometimes these earlier experience can hinder correct understanding of new concepts. Misconceptions about variables can arise from previous experiences in algebra where a *variable* is a letter replacing a value in an equation to be solved (Ma, Ferguson, Roper, & Wood, 2011). For example, `a = 6; a = b + 4`, is than expected to solve the equation for b, `b = 2`. Or the equal-sign is conceived to make both sides equal, so a value can also be moved from a variable left from the equal-sign to a variable on the right (opposed to assignment in programming that is always only done from right to left). For example, `a = 4; b = 3; b = a`, can result in either (correctly) `a = 4` and `b = 4`, or (incorrectly) `a = 3` and `b = 3`.

Inter-human communication experiences can also cause wrong expectations of variables. In everyday communication we learn that contextual information supports correct understanding. Even when being imprecise humans can interpret the meaning of words like `smallest`. A novice programmer may, erroneously, expect the computer as well to understand context and intention (Pea, 1986; Pea, Soloway, & Spohrer, 1987).

Another potential source of misconceptions is that of the container analogy; a variable is like a box (Smith et al., 1993; Ben-Ari, 1998). This analogy can help explain that a variable is given a name and can hold a value. However, the analogy may also result in students to think that a variable can contain more than one value, or that a value is removed when assigned to another variable.

Lastly, there have been identified various misunderstandings in the semantics of assignment statements such as assumptions that variables are swapped or added (Ma, Ferguson, Roper, & Wood, 2007).

In previous work (Plass, 2015) we extensively reported misconceptions identified from literature[3]. These misconceptions, grouped by origin category, are listed in Table 2.1.

Table 2.1: Identified misconceptions in variable assignment for primitive types in imperative programming.

Mathematics	Human interaction	Container analogy	Semantics
M1 - Variables are set to being equal, also from left to right.	H1 - Variables cannot contain values in conflict with their name.	C1 - A value is moved, a variable on the right side loses the value it contained.	S1 - Values are tested for being equal, which is true or false.
M2 - The statement is an equation to be solved.	H2 - Variables contain values that make sense given their name, but were never explicitly assigned.	C2 - Variables can contain multiple values, like a box can contain multiple items.	S2 - The receiving variable is on the right side.
M3 - Variables are fixed values or constants, assigned a value once.			S3 - The values of the variables are swapped.
			S4 - The new value is added to the previous value.
			S5 - Results can only be stored in variables not mentioned in the expression on the right side.

3 Additional misconceptions have been identified from the results of our study. These misconceptions are outside the scope of this report since the intervention was not designed to address these —then unidentified— misconceptions. Interested readers are referred to this work (Plass, 2015).

2.2.2 Assessing Misconceptions

To design reforming instructions, we need to detect mistakes *and* understand the underlying non-viable model (Herman et al., 2008). Misconceptions held by students are often assessed with think-aloud protocols and task-based interviews, which give insight into thoughts but also influence thinking. Alternatively, misconceptions can be assessed with a directed test.

Common think-aloud approaches to uncover misconceptions include asking students to explain what they think happens in particular code segments (Bayman & Mayer, 1983; Kurland & Pea, 1985; Pea et al., 1987), giving small problems with code segments to solve while letting the student think out loud and asking about specific concepts (Kaczmarczyk, Petrick, East, & Herman, 2010), or asking open ended questions (Tew, 2010). Although these approaches may reveal misconception, they can change the sequence of thinking or slow down the process (Hickman & Monaghan, 1993). A partial solution can be found in the use of a smartpen (e.g., Livescribe[4]). The pen records writing actions and audio, allowing the student to work on their normal pace and reflect asynchronous upon their work.

Attempts to develop a formal test assessing students understanding of programming concepts —including, but not limited to variables— have been undertaken. In the FCS1 (Tew & Guzdial, 2011) multiple choice test each incorrect answer indicates a specific misconception. However, the test contains only three items about variables, and the possible incorrect answers were not constructed from misconceptions but rather created based on guidelines (Miller, Linn, & Gronlund, 2009, pp. 194–217). Moreover, the FCS1, is not available for general use (Taylor et al., 2014). Dehnadi (2006) developed a test focusing on assignment of variables of primitive types. This test consists of multiple choice questions based on code fragments. Answers have been mapped to behavioural mistakes, but not to the misconception underlying these mistakes.

Misconception Assessment Test

Based on the work of Dehnadi (2006), a directed test assessing misconceptions held by a student was developed and presented in Plass (2015), with some alterations in construction and interpretation of the answers. Our test use open-ended questions to avoid response bias and because we assumed the list of identified misconceptions to be incomplete. Further, we mapped incorrect answers to identified misconceptions rather than behavioural mistakes because we were interested in the non-viable models underlying the mistake. For all identified misconceptions,

[4] `http://www.livescribe.com/nl/smartpen/`

we constructed programming code snippet(s) eliciting certain incorrect responses whenever a student holds a certain misconception. For all programming code snippets the student has to answer the values for all variables after execution of the code; predicted incorrect responses have been mapped to misconceptions. The resulting assessment test is available in Appendix B, the programming code snippets and questions mapped to misconceptions are given in Table 2.2.

The assessment test was carried out with novice programmers, and showed able to detect misconception $\mathrm{H2}$, $\mathrm{M1}$, $\mathrm{M2}$, $\mathrm{S1}$ $\mathrm{S2}$, and misconception $\mathrm{S4}$. Conversely, no evidence was found that the test was able to detect misconception $\mathrm{H1}$, $\mathrm{C1}$, $\mathrm{C2}$, $\mathrm{M3}$, $\mathrm{S3}$, and misconception $\mathrm{S5}$.

Some identified misconceptions were not ($\mathrm{C1}$) or hardly ($\mathrm{H1}$, $\mathrm{C2}$, $\mathrm{S3}$) detected in the sample (see Appendix D, Section D.2). This may be due to ineffective assessment or participants not holding these misconceptions. Though, some identified misconceptions clearly could not be detected by the test due to limitations in the constructed programming code snippets. First, on the basis of the expected incorrect responses differentiation between misconception $\mathrm{S1}$ and misconception $\mathrm{M3}$ was impossible. For example, for code snippet `pre_h`, `Dim a As Integer; Dim b As Integer; a = 4; b = 3; b = a`, the incorrect response to question `pre_h2`, `b = 3`, (combined with the correct value for `pre_h1`, `a = 4`) matches the expected values mapped to both misconception $\mathrm{S1}$ and $\mathrm{M3}$ as presented in Table 2.2. In few cases however, misconception $\mathrm{S1}$ could be uniquely detected by unanticipated responses such as "`Error, not equal`", indicating that the participants believes equality need to be tested. In a similar vein, misconception $\mathrm{S5}$ elicited the same incorrect responses as misconception $\mathrm{M2}$, but, unique detection of misconception $\mathrm{M2}$ was possible based on code snippets `pre_j` and `pre_k` (see Table 2.2).

Further, unanticipated mistakes were observed resulting in identification of additional misconceptions, namely, misconception $\mathrm{O1}$ that the value for one variable is computed and the other is set to be equal, misconception $\mathrm{O2}$ that a known value of another variable is used when there is no value explicitly assigned, and three sub-implementations of misconception $\mathrm{M2}$ that a statement is an equation to be solved. These additional misconception have been identified after the data collection and therefore are not included in the present study.

Table 2.2: (pre)Test questions mapped to identified misconceptions. Answers that do not differentiate between correct values and misconceptions are not listed, or depicted in grey if they support misconception detection by another question. Coloured cells mark questions designed to assess the specific misconception. ∗ marks answers uniquely detecting a misconception. † marks answers that combined with the other questions of the code snippet detect a misconception. All code and values are for Visual Basic.

Code	Question	Variable	Correct	H1 - Variables cannot contain values in conflict with their name	H2 - Variables contain values that were never explicitly assigned	M1 - Variables are set to being equal, also from left to right	M2 - The statement is an equation to be solved	M3 - Variables are fixed values or constants	C1 - A value is moved	C2 - Variables can contain multiple values	S1 - Variables are tested for equality	S2 - The receiving variable is on the right side	S3 - The values are swapped	S4 - The new value is added to the old value	S5 - Results can only be stored in variables not in the expression
`Dim tien As Integer`	a	tien	0		10*										
`Dim dozijn As Integer`	b	dozijn	0		12*										
`Dim drie As Integer` `drie = 5`	c	drie	5	0 * Error	3*										
`Dim straatnaam As Integer` `straatnaam = 101`	d	straatnaam	101	0 * Error											
`Dim groot As Integer` `Dim klein As Integer` `groot = 10` `klein = 20` `groot = klein`	e1	klein	20	10	10	10 or 20	1*	20	no value*	20	20	10†	10	20	
	e2	groot	20	Error	20	10 or 20	2*	10	20	10, 20*	10	10†	20	30*	
`Dim Hugo As Integer` `Dim Tim As Integer` `Hugo = 12` `Tim = Hugo + 3`	f1	Hugo	12	Error no value					no value†		12				
	f2	Tim	15	Error no value					15		0*				
`Dim a As Integer` `Dim b As Integer` `a = 7` `b = a`	g1	a	7			0 or 7			no value*		7	0†	0		
	g2	b	7			0 or 7			7		0†	0†	7		
`Dim a As Integer` `Dim b As Integer` `a = 4` `b = 3` `b = a`	h1	a	4			3 or 4	-1*	4	no value*	4	4	3†	3†	4	
	h2	b	4			3 or 4	1*	3	4	4, 3*	3	3†	4	7*	
`Dim x As Integer` `Dim y As Integer` `x = 10` `y = 20` `x = y`	i1	x	20			10 or 20	2*	10	20	10, 20*	10	10†	20	30†	
	i2	y	20			10 or 20	1*	20	no value*	20	20	10†	10†	20	
`Dim x As Integer` `Dim y As Integer` `y = 8` `y = x + 10`	j1	x	0				-2*	0		0	0			0	
	j2	y	10				8†	8		8, 10*	8			18*	
`Dim a As Integer` `Dim b As Integer` `a = 8` `a = b * 4`	k1	a	0				8†	8		8, 0*	8			8	
	k2	b	0				2*	0		0	0			0	
`Dim i As Integer` `i = 1` `i = i + 1`	l	i	2			0 or 1 or 2*	Error	1		1, 2*	1			3*	1 Error
`Dim a As Integer` `Dim b As Integer` `a = 6` `b = a + 1`	m1	a	6						no value*		6		0*		
	m2	b	7						7		0*		7		
`Dim x As Integer` `Dim y As Integer` `x = 8` `y = x`	n1	x	8						no value*		8		0*		
	n2	y	8						8		0*		8		
`Dim a As Integer` `Dim b As Integer` `Dim c As Integer` `a = 10` `b = 20` `c = 30` `a = b` `c = a`	o1	a	20			10 or 20 or 30	2*	10	no value*	10, 20*	10		30†	30†	
	o2	b	20			10 or 20	0*	20	no value*	20	20		10*	20	
	o3	c	20			10 or 30	3*	30	20	30, 10, 20*	30		20	60*	
`Dim a As Integer` `Dim b As Integer` `a = 10` `b = 20` `a = b` `b = a`	p1	a	20			10 or 20	0*	10	no value*	10, 20*	10	10†	10	30*	
	p2	b	20			10 or 20	2*	20	20	20, 10*	20	10*	20	50*	

2.3 Video Instruction

The number of available video materials for educational purposes have been rapidly increasing, amongst other reasons due to relatively cheap, easy access to technologies and easy distribution via internet. Educational videos have advantages over paper tutorials or classroom instructions, such as the richness of the representation and independence of time and location, but the quality of productions is not necessarily sufficient for them to be (effective) educational videos (Winslett, 2014; Van der Meij & Van der Meij, 2013; Höffler & Leutner, 2007).

Instructional design is subject of academic interest. This work, however, is often theoretical and provides no design guidelines. For example, Winslett (2014) reviewed publications on educational videos and classified production types and learning objectives, but did not attempt to provide best practices or principles. Even so, Wilson (2013) presented different views on instructional design and its possibilities, but did not specify a set of instructional principles or strategies, even tough, the author stated that instructional design theories differ from educational psychology and learning sciences by its aim to guide teachers in the process of creating courses and media. The author's concluding thoughts are that "Instructional design practice involves knowledge production just as doing research does. Art, science, craft, analysis, testing —all have a role for designing good instruction[. . .]." Höffler and Leutner (2007) presented a detailed theoretical framework, and give some suggestions on how this influences learning. For example, they state that learning outcomes are improved by presentation of both verbal and visual information —the "multimedia principle" (Mayer, 2002). From their meta-analysis the authors concluded that animations out perform static images, and that representational animations are superior to animation with a decorational function.

An online tutorial for instruction videos (Soofos, 2015) states that there are three phases while making a video: preparation, filming and editing. The preparation phase includes a concise planning of the video content, consideration of the goal and integration with other course materials, planning of interactive elements, writing the scenario and voice-over, read and rewrite removing redundancy, jargon and overcomplexity, and lastly, gathering of materials. Recording should always be done multiple times while paying attention to details such as presentation and lightning. For editing their advise is to adhere to the "less is more" principle, and keeping the video as short as possible or segment the video to maintain engagement. During editing voice-over and visual elements should get aligned, effects should only be added if they enhance the production. Finally, the video should be published on a platform accessible from various devices. However, this online tutorial seems to be based on common senses rather than on explicit instructional design.

Van der Meij and Van der Meij (2013) developed concrete guidelines (see Section 2.3) for the design of instructional videos for software training, based on research about information processing and instructional design principles. The authors found that, videos adhering to these guidelines outperformed paper tutorials in improvement of both skill and motivation. However, these guidelines were developed for tutorial videos (i.e., procedural knowledge) and not all guidelines may be equally important for other learning objectives. Clearly, for transfer of conceptual knowledge is deemed irrelevant guideline 5, which explicitly states to provide procedural rather than conceptual information. Further, additional practices may be needed for a production to serve conceptual learning objectives.

Guidelines for Tutorial Videos **by Van der Meij and Van der Meij (2013)**

Guideline 1: Provide easy access Make the video easy to find.

- **Guideline 1.1 Craft the Title carefully** Use a verb and object to indicate what the video is about, and avoid use of jargon in introductory materials.

Guideline 2: Use animation with narration Display a sequence of events that expresses an actual scenario of use.

- **Guideline 2.1 Be faithful to the actual interface in the animation** The content and format should correspond (congruency principle); demonstrate task execution in context.
- **Guideline 2.2 Use a spoken human voice for the narration** Explain what is happening on the screen; learning is enhanced by provision of both visual and auditory sensory information (modality principle).
- **Guideline 2.3 Action and voice must be in synch** Simultaneous presentation is more effective than successive (temporal contiguity principle).

Guideline 3: Enable functional interactivity Unfold the scenario fit to the users capabilities; use interaction to pause the stream of information (limited capacity model).

- **Guideline 3.1 Pace the video carefully** Keep a conversational tempo, speaking to quickly leads to overload, to slow to boredom.
- **Guideline 3.2 Enable user control** Enable standard media player controls to allow the user to (re)inspect and focus. Use segmentation to activate the user.

Guideline 4: Preview the task Give an outline and clarify main goals.

- **Guideline 4.1 Promote the task** Use before and after displays to clarify task relevance.

Guideline 4.2 use a conversational style to enhance perceptions of task relevance A personal style is more appealing that a formal style, thereby stimulating active processing (personalization principle). Further, a familiar style requires less cognitive effort.

Guideline 4.3 Introduce new concepts by showing their use in context Introduce vocabulary and explain concepts when relevant (i.e., appearing during the demonstration) to reduce working memory load.

Guideline 5: Provide procedural rather than conceptual information Focus on the learning objective.

Guideline 6: Make tasks clear and simple Use easy to understand, concrete and realistic explanations and eave out all non-essential information (coherence principle).

Guideline 6.1 Follow the user's mental plan in describing an action sequence Follow the sequence of actual task execution.

Guideline 6.2 Draw attention to the connection of user actions and system reactions Highlight the relation between action and system response.

Guideline 6.3 Use highlighting to guide attention Information that belongs together should be presented in close proximity (spatial contiguity principle). Highlight feature that require attention and distinguish highlights from the actual interface (signalling principle).

Guideline 7: Keep videos short A video should last 1 to 3 minutes. Create meaningful segments with a clear beginning and end to optimize engagement and minimize cognitive load (segmenting principle).

Guideline 8: Strengthen demonstration with practice During instruction the problem and solution are explained, during practice the user actively solved the problem on their own. Practice consolidates and enhances learning, and serves as self-assessment for the user.

Chapter 3

Intervention

Our aim was to develop material to instruct correct conceptual knowledge about variables of primitive types in imperative programming, thereby, addressing any misconceptions a student may hold on this topic. Therefore, we created an interactive, educational video. First, we defined the learning goals addressing identified misconceptions. Next, following the constructivist approach, we ordered these learning goals by increasing complexity. Then, we created a storyline naturally appointing each learning goal. Following the story of hacker Nelly trying to retrieve a passcode, a few lines of code are traced, gradually explaining assignment statements. The video is paused at strategical moments to maintain or regain attention and assess knowledge reception. The scenario script, including questions, is available in Appendix C, and the video is available on YouTube (`https://youtu.be/WARZCZ_D66Y`).

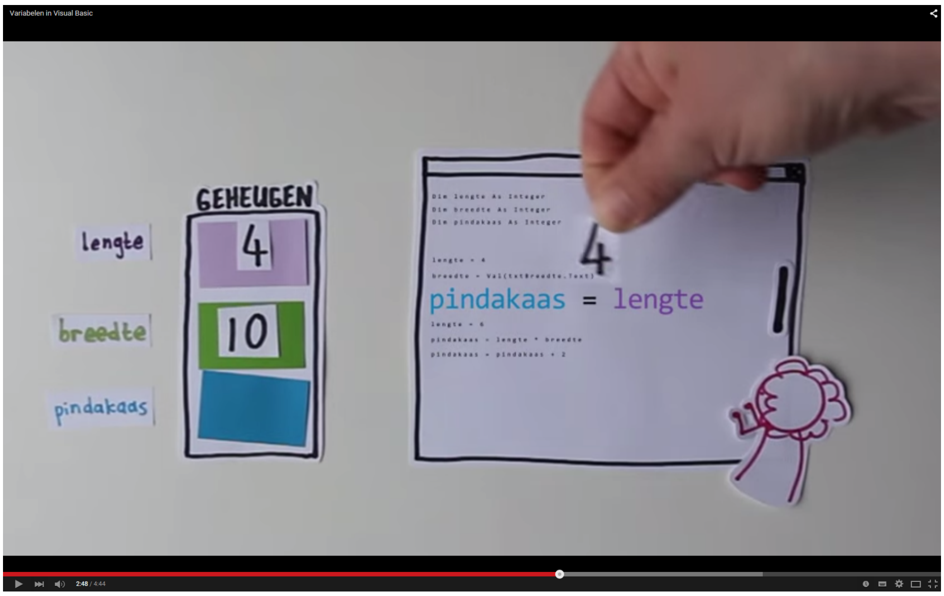


Figure 3.1: still of the created video.

3.1 Learning Objectives

The objective of the video instruction is teach viable models about variables by providing alternative knowledge addressing identified misconceptions students may hold. Based on the misconceptions from the literature and presented in Section 2.2.1, we formulated six learning goals as presented below. The relations between these learning goals and identified misconceptions are presented in Table 3.1. These learning goals are one by one instructed in the video, showing both the programming code and a visualisation of the computer memory.

learning goal 1 The name of a variable is of no meaning.

learning goal 2 A value is assigned to a variable, using the name of the variable followed by the =-symbol and the expression for the value (e.g., `a = 1`).

learning goal 3 A value is copied to a variable (i.e., for `b = a`, both a and b have the same value [not the same object]).

learning goal 4 The value of a variable can be changed at runtime.

learning goal 5 A variable can store one (1) value, on change the previous value is lost.

learning goal 6 The value of a variable can be changed as the result of a computation using the value stored in this variable (e.g., `a = a + 1`).

Based on the programming course progress and student performance, we assumed prior knowledge listed below. Nonetheless, these items are briefly touched upon at the start of the video to refresh memory and provide a comfortable start.

A Variables are used to improve programming code efficiency, readability and flexibility.

B Each variable has a name, which is a referent to reserved space in the computer memory.

C Each variable has a value, which is stored in and read from the memory using the variable name. (In Visual Basicif no value is assigned a default value is given.)

D In Visual Basic, a variable is of a specific data type, which is given in the declaration statement after `As`.

   a Variables of Integer type can only contain non-decimal numeric values.

E In Visual Basic, when you do not specify an initial value a default value is assigned, which is `0` for variables of a numeric type.

Table 3.1: Learning goals related to misconceptions. Extended version of Table 2.1, indication misconceptions and learning goals defined to address these misconceptions.

Learning goals	Mathematics	Human interaction	Container analogy	Semantics
1 The name of a variable is of no meaning.		H1 - Variables cannot contain values in conflict with their name. H2 - Variables contain values that make sense given their name, but were never explicitly assigned.		
2 A value is assigned to a variable, using the name of the variable followed by the = symbol and the expression for the value.	M1 - Variables are set to being equal, also from left to right. M2 - The statement is an equation to be solved.			S1 - Values are tested for being equal, which is true or false. S2 - The receiving variable is on the right side.
3 A value is copied to a variable.			C1 - A value is moved, a variable on the right side loses the value it contained.	S3 - The values of the variables are swapped.
4 The value of a variable can be changed at runtime.	M3 - Variables are fixed values or constants, assigned a value once.			
5 A variable can store one (1) value, on change the previous value is lost.			C2 - Variables can contain multiple values, like a box can contain multiple items.	S4 - The new value is added to the previous value.
6 The value of a variable can be changed as the result of an expression using the value stored in this variable.				S5 - Results can only be stored in variables not mentioned in the expression on the right side.

3.2 Instruction Material Design

For the instruction material a video was chosen because this type of media is familiar to and popular amongst our target audience, and video instructions provide high motivation and support long-term knowledge acquiring (van der Meij & van der Meij, 2015; Höffler & Leutner, 2007). For the design of the instructional video we relayed, as advised by Wilson (2013), on all three instructional design approaches: 1) Artistic, using intuition (e.g., storyline); 2) Emperical, collecting tryout data (e.g., topic selection); and 3) Analytical, using principles from theory (see Chapter 2).

The chose video style is a paper-cut stop-motion style due to its relatively easy, cheap and flexible production possibilities. Moreover, the style is a commonly appreciated animation technique. One of the advantages of the production style was that it allowed us to create three versions for different programming languages[1] whit exactly the same animations and voice-over. The story-line of hacker Nelly was added to make the scenario more playful and interesting, and provide some context for otherwise seemingly meaningless lines of code.

Throughout the video is visualised a simulation of code execution with a voice-over explaining step-by-step the effect on the variables and their values in the computer memory. This presentation is consistent with recommendation for conceptual knowledge transfer given by Van der Meij and Van der Meij (2013) and recommendations for instructing variables by Kuittinen and Sajaniemi (2004). After a short introduction, presenting assumed prior knowledge, the scenario gradually builds up knowledge about variables , including all learning goals. As advised by Kuittinen and Sajaniemi (2004), starting with labels for variables, followed by assignment of a numeric value and than advancing to more complex assignment statements.

The planned scenario resulted in a relatively long video (4:44, while 1–3 minutes was recommended (Van der Meij & Van der Meij, 2013)). Therefore, pauses were included at logical moments (i.e., after instructing each of the learning goals) to reactivate viewer attention. Mayer (1988) suggested these pauses to improve effectiveness, as well was this recommended in guidelines by Van der Meij and Van der Meij (2013). Each pause included a multiple choice question, usable for self-assessment of correct understanding, and facilitate viewer control of pace and segment replay. This optimises viewer understanding before proceeding to the next item, which supports the constructivist approach of building upon prior knowledge (Ben-Ari, 1998; Kuittinen & Sajaniemi, 2004).

[1]The study was planned to be conducted with participants following a programming course in either Visual Basic, Java or JavaScript. Therefore, the animation and voice-over were kept language independent, and language specific syntax was added in a separate animation layer that was different for each of the three version of the video.

3.2.1 Adherence to Guidelines

Although design decisions initially may originate from intuition or prior experience, they do adhere to the guidelines proposed by Van der Meij and Van der Meij (2013) as presented in Section 2.3. When applicable the scenario and production were advanced by explicit application of these guidelines.

Although the guidelines were designed for instruction of procedural knowledge, they mostly apply as well to instructions of conceptual knowledge. Except for **guideline 5**, which explicitly states to provide procedural rather than conceptual information, while expertise programming skills require conceptual knowledge rather that procedural (Ben-Ari, 1998). Acknowledging this, Van der Meij and Van der Meij (2013) provide an alternative; a simulation displaying problem-solving steps in combination with a voice-over informing about the rationale behind each step. This description seamlessly fits the design of our video.

Guideline 1 The video has a short but descriptive title "*How do variables work?*" The video was published online and a hyperlink was offered via the schools prevalent media platform accessible to all targeted students.

Guideline 2 The video starts with a quick recap of what variables are used for and then frames the main questions of how variables work. Each step is visualised and synchronously explained by a (human) narrator. The animation does not show the actual interface to improve readability, but the congruency principle is applied by showing compatible, working programming code. The computer memory bank with the variables is a conceptual representation.

Guideline 3 The video follows a slow conversational pace and is segmented into six parts corresponding with the learning goals. The Hapyak video player provides basic user control (play, pause, replay, and skipping), and shows segments in a timeline. Segments were created by pauses with multiple choice questions. The video design consciously deviated from **guideline 7**, but instead used these segments to channel attention and reduce cognitive load.

Guideline 4 The aim of the video is clearly stated and the scenario is introduced. Then, assignment statements are instructed with step-wise increasing complexity. New information is embedded into the story-line, visualised (in code and memory), and explained by the narrator.

Guideline 6 Statements and explanations were kept as short and simple as possible, and jargon was avoided in the explanations. Sequencing of steps started with the most basic expression and gradually increased in complexity. The memory visualisation showed the relation between programming code (user actions) and the effect on the system (system reactions). The current line of code was highlighted, colouring variable names matching the memory visualisation and increasing the font size, improving readability.

Guideline 8 For the purpose of the present study the video was not embedded into a course design and no additional exercises were added. However, the video was offered at a logical moment during the course when students could immediately apply the knowledge from the video in the course assignments.

Chapter 4

Method

We conducted a quasi-experimental, within-subject empirical study to evaluate an instruction material designed to teach correct understanding of variables in three classes with students of secondary education enrolled in a introductory Visual Basic programming course. Students watched an interactive video as described in Chapter 3. Misconceptions held about variables were measured prior to and after watching this video.

4.1 Participants

A total of thirty-seven Dutch students, divided over three classes, in their fourth year of secondary education were recruited to participate in the study[1]. Students were following general higher education (havo, 2 classes) or pre-university education (vwo, 1 class), enrolled in the computer science curriculum, including an introductory programming course, and obliged to participate during class. Students scoring above 60% in the post-test were rewarded a small benefit at their final exam for the programming course. Four students did not complete the test(s) and have been excluded from further analysis. The remaining 33 participants (27 male, 6 female, 15-18 y.o.) followed higher general secondary education (havo, $n = 22$) or pre-university education (vwo, $n = 11$). All participants did have some prior experience in HTML and CSS. All but one reported no further programming experience.

[1]The study was also conducted at another school of secondary education and a university of applied science, but data obtained from these samples were exclude from further analysis due to low number of participants (respectively 2 and 6), difference in programming language (slightly alternating the answer model), and (unsupervised) procedure.

4.2 Materials

The study materials are the pre-test to detect initial misconceptions, the post-test to measure the learning effect of the instruction video, and the interactive video to instruct correct understanding of assignment statements for variables in imperative programming. Explanation of the procedure and hyperlinks to the assessment tests and video were provided through the school's learning environment[2] (see Figure 4.1). All materials were accessed trough a desktop computer equipped with in-ear headphones.

Pre- and Post-Test were developed to assess whether a student has a correct understanding of or holds misconceptions about variables based on the work of the work of Ma et al. (2007); Dehnadi and Bornat (2006), as described in Section 2.2.2. The pre-test contained, next to the assessment questions, two questions asking for level of education, and prior programming experience. The 29 assessment questions ask for the value of each variable after execution of a given programming code snippet. For each question, incorrect responses were predicted and mapped to specific misconceptions, and for each misconception at least two programming code snippets (with one or more questions) were constructed to detect the specific misconception (see Table 2.2). The question order was randomised to minimise the learning effect of the test itself. The post-test contained a variation of each pre-test code snippets and assessment questions, slightly altering names and values to ensure assessing misconceptions in a similar way. The assessment tests were published via Socrative[3]. A representative screen and all pre- and post-test code snippets and questions are presented in Appendix B.

Interactive Video was developed to instruct correct understanding of the learning goals addressing identified misconceptions about variables as presented in Table 3.1. The video loosely follows a constructivist approach by giving context and gradually building up knowledge while triggering active involvement. Further, if applicable, we adhered to the guidelines for design of an instruction video by Van der Meij and Van der Meij (2013). A paper-cut, stop-motion animation follows hacker Nelly while she is tracing a few lines of code to retrieve a hidden value —the password of architect Nico. For each line of code, the effect of statement execution on the variables and their values in the computer memory is visualised, and a voice-over explains what happens in natural language. The resulting video has been published

[2] `https://candea.itslearning.com/`

[3] `https://www.socrative.com/`

on YouTube[4]. Six required multiple choice questions were included to retain attention, and highlighting important learning goals. These questions were added using the freely available web based tool Hapyak[5]. The script, questions, and video stills are available in Appendix C. Details on the design rational are given in Chapter 3.

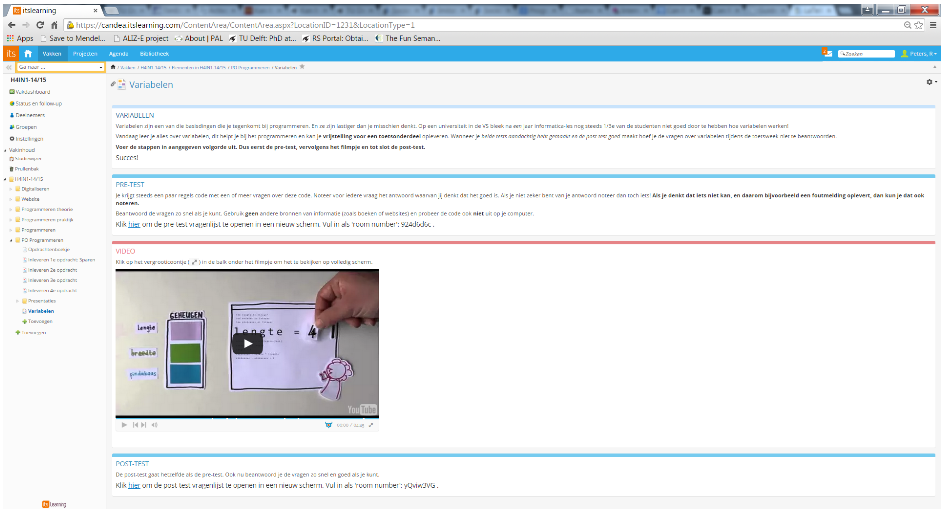


Figure 4.1: illustration of the format in which the instructions and hyperlinks to the pre-test, interactive video and post-test were presented to the participants via ItsLearning. (Readable instructions are available in Appendix C, Section C.3)

4.3 Procedure

Participants were seated at a computer in the classroom. They received a <1 minute oral instruction emphasising that it was not allowed to collaborate or look up answers, and directing the participants to further instructions at the learning environment. Hereafter, the participants started their computer and visited the directed learning environment page. From this page they navigated to the pre-test and subsequently the interactive video and afterwards the post-test. Although a time limitation was not explicitly given, all students finished within 45 minutes, before class ended. The study was fully conducted under supervision in the classroom. Students were allowed to ask procedural questions only (e.g., problems with headphones), substantive questions about the material were not answered. This procedure was repeated three times, once for each participating class.

[4] https://youtu.be/WARZCZ_D66Y
[5] http://corp.hapyak.com/

4.4 Measures

Misconceptions held by each participant were assessed before and after watching the interactive video by the assessment questions of the pre- and post-test. In the post-test one question was accidentally left out, leaving 28 questions to be used for evaluation of the intervention; pre-test questions `3--22` and `24--31`, and post-test questions `1--28`. Participants level of education and prior programming experience were measured by the first two pre-test questions. The responses to a total of 2 demographic and 56 assessment questions were recorded for 33 participants. The obtained data is available in Appendix D, Section D.1.

We applied some transformations to the assessment questions and response data before analysis. The questions were ordered aligning pre- and post-test questions constructed to assess the same misconceptions in correspondence with Table 2.2, and renumbered alphanumerical [`a--p`] for each programming code snippet and numeric [`1--3`] for each question about the code snippet, with a prefix [`pre or post`] indicating the test. The responses were recoded rating the possibly held misconception. For non-differential responses, combined misconceptions were rated. Further, due to the frequency of responses indicating a lack of assumed prior knowledge these items were added to the categorical values. Finally, we added an *other* category to rate remaining incorrect responses. Resulting in a list of 23 possible values as given in Table 4.1. Following, the scores for each category was calculated, counting for each participant the number of times a category was rated. Details on the recoding and transformation process are given in the remainder of this section.

The resulting dataset includes three independent variables (participant, educational level, and prior programming experience), a total of 56 categorical variables —28 explanatory and 28 outcome— (the recoded pre- and post-test responses) with 23 possible values (0 for a correct response and for incorrect responses an abbreviation of the misconception, a letter of prior knowledge item, or other), and 34 ordinal variables (calculated pre- and post-test scores for specific clusters of categories i.e., all, class or error, identified misconceptions, and learning goals) —17 explanatory and 17 outcome. This dataset in available in Appendix D, Section D.2, and was used for analysis evaluating the effectiveness of the video in addressing misconceptions.

4.4.1 Recoding

First, we recoded the responses rating the misconception detected from predicted values as listed in Table 2.2. In case a misconception could only be uniquely detected by multiple responses related to one programming code snippet we looked at all responses, but rated the misconception only for incorrect responses. This

Table 4.1: Categorical values. Coding schema for rating of categories, including the abbreviation of a misconception, the letter of an assumed prior knowledge item, or other.

Value	Label
0	Correct answer
-1	Other (unidentified) mistake
A	Indicating lack of prior knowledge A (variables improve code flexibility, efficiency and readability)
B	Indicating lack of prior knowledge B (variables have a name, which refers to reserved memory space)
C	Indicating lack of prior knowledge C (variable has value)
D	Indicating lack of prior knowledge$\mathrm{D}$ (variables of type Integer can contain only non-decimal numbers)
E	Indicating lack of prior knowledge E (the default value for Integer variables is 0)
$\mathrm{H1}$	Detected misconception $\mathrm{H1}$ (variable cannot contain value in conflict with name)
$\mathrm{H2}$	Detected misconception $\mathrm{H2}$ (variables contain values never explicitly assigned but logical given the name)
$\mathrm{M1}$	Detected misconception $\mathrm{M1}$ (variables are set to be equal, also from left to right)
$\mathrm{M2}$	Detected misconception $\mathrm{M2}$ (a statement is an equation to solve)
$\mathrm{M3}$	Detected misconception $\mathrm{M3}$ (variables are fixed values or constants)
$\mathrm{C1}$	Detected misconception $\mathrm{C1}$ (a value is moved)
$\mathrm{C2}$	Detected misconception $\mathrm{C2}$ (variables can contain multiple values)
$\mathrm{S1}$	Detected misconception $\mathrm{S1}$ (variables are tested for equality)
$\mathrm{S2}$	Detected misconception $\mathrm{S2}$ (the receiving variable is on the right side)
$\mathrm{S3}$	Detected misconception $\mathrm{S3}$ (the values are swapped)
$\mathrm{S4}$	Detected misconception $\mathrm{S4}$ (the new value is added to the previous)
$\mathrm{S5}$	Detected misconception $\mathrm{S5}$ (results can only be stored in variables not mentioned in the expression)
$\mathrm{H1S1}$	Detected misconception $\mathrm{H1}$ (name conflict) or misconception $\mathrm{S1}$ (test for equality)
$\mathrm{H2S3}$	Detected misconception $\mathrm{H2}$ (values logical to name) or misconception $\mathrm{S3}$ (swapped)
$\mathrm{M2S5}$	Detected misconception $\mathrm{M2}$ (equation to solve) or misconception $\mathrm{S5}$ (target var cannon be in expression)
$\mathrm{M3S1}$	Detected misconception $\mathrm{M3}$ (variables are fixed values) or misconception $\mathrm{S1}$ (variables are tested for equality)

was done to optimise the accuracy of misconception detection without changing the rating frequency. Meaning that if a programming code snippet had two questions and one was answered correctly and the other incorrect, the correct answer may be used to understand the misconception underlying the incorrect answer, but the misconception was rated only once. While, if both questions were answered incorrect, the misconception was rated twice. For example, take programming code snippet `Dim a As Integer; Dim b As Integer; a = 7; b = a`, and question (pre_g1) "What is the value for `a`?" The incorrect response `a = 0` would be rated S3 when the correct value (`b = 7`) was given, while S2 was rated for both questions when the (incorrect) response to pre_g2 was `b = 0`. However, if the value for `b` was `0`, but for `a` correct (7), then (only) question pre_g2 was rated S1.

A new category was created when incorrect responses did not differentiate between multiple misconceptions, and neither could be disproved. For example, take code snippet `pre_i Dim x As Integer; Dim y As Integer; x = 10, y = 20; y = y`. The combined incorrect responses `x = 10` and `y = 20` were the predicted values for misconception M3 as well as misconception S1, thus these incorrect responses were rated M3S1. Based on non-differential values, the following concatenated categories have been identified: H1S1, H2S3, M2S5, and M3S1.

Then, remaining incorrect responses —not corresponding to any of the predicted values, mapped to identified misconceptions— but clearly indicating a lack of assumed prior knowledge as listed in Section 3.1 were rated accordingly. Meaning that, all responses that included a variable name or an expression were rated C (a variable has a value), non-integer values were rated D (a variable of data type Integer cannot contain decimal numbers), and `no value` (and a few occasions of `Error`) were rated E (variables of type Integer get default value `0` if no value is explicitly assigned). Assessing misconceptions was prevalent, thus lack of prior knowledge was only rated if the response did not match to any of predicted values detecting misconceptions. Except for cases were the misconception would result in an incompatible type. For example, take programming code snippet `pre_e Dim groot As Integer; Dim klein As Integer; groot = 10; klein = 20; groot = klein`, and question (pre_e1) "What is the value for `klein`?" The incorrect response `0.5` was rated D rather than M2 because a student with proper assumed prior knowledge should have know that this value is not possible and rather responded with `Error` or round-up number `1`.

Further, a catch-all category was included rating all remaining incorrect responses that did not match the predicted values detecting identified misconceptions, nor clearly indicated a lack of assumed prior knowledge. These unanticipated behavioural mistakes could originate from different issues. Sometimes a participant appeared to be clueless on the value of the variable resulting in non-sense making responses such as “variable”, “user input”, or “`563556`”. However, often there might be a combination of misconceptions underlying these mistakes. The fast majority of these unclassified incorrect responses included a numeric value that may have resulted from an algebraic solution. Although meaningless at first sight, there is often ‘logic’ behind these responses when including some *magical value contagion*. For example, take code snippet `Dim a As Integer, Dim b As Integer, a = 10, a = b * 8`, question (`post_k1`) “What is the value for `a`?”, and response `80`. When ‘magically transferring’ the old value of `a` (`10`) to `b`, it results in a perfect equation; $80 = 10 \times 8$. This may indicate holding misconception $\mathrm{M2}$, but also suggests misconception $\mathrm{S3}$. There were too many, inconsistent, unanticipated behavioural mistakes to draw solid conclusions about the assumptions underlying these mistakes. Moreover, identifying misconceptions is outside the scope of this report, for the remainder of this work these are all categorised as *other*.

4.4.2 Transformation

Finally, we scored per participant per test the total number of incorrect responses, the number of incorrect responses per class (i.e., identified misconception, prior knowledge, or other) and the number of ratings per misconception, the score per learning goal was calculated based on the ratings of related misconceptions. Thereby we calculated scores on four levels: overall, class of error, misconceptions and for learning goals, with respectively 7 and 5 clusters for last two (see Table 4.2).

The overall score was obtained by calculating all incorrect responses. Class of error scores were obtained by counting all the possible categorical values detecting identified misconceptions, or indicating assumed prior knowledge or the number of *other* incorrect responses.

Misconception scores were obtained by counting all ratings of possible categorical values including each specific misconception. Minimally including the misconception abbreviation itself, and possibly concatenated abbreviations. Seldom reported ratings for non-differential values were counted with the predominant misconception. Often rated $\mathrm{M3S1}$ was included in the calculation of the scores for both misconception $\mathrm{S1}$ and $\mathrm{M3}$. Noteworthy, misconception $\mathrm{M3}$ could never be uniquely detected based on the tests, so the score of misconception $\mathrm{M3}$ was based only on ratings of $\mathrm{M3S1}$. Misconceptions $\mathrm{H1}$, $\mathrm{C1}$, $\mathrm{C2}$, $\mathrm{S3}$, and $\mathrm{S5}$ have been rejected because these were not or hardly present in our sample.

Learning goal scores were obtained by counting all ratings of possible categorical values for all misconceptions related to a specific goal. For example, scores for **learning goal 2** are based on the number of ratings of misconception $\mathrm{M1}$, $\mathrm{M2}$, $\mathrm{S1}$, and $\mathrm{S2}$, and also non-differential ratings $\mathrm{H1S1}$ and $\mathrm{M2S5}$.

We did choose to *not* normalise the scores because choice of the divider would be prone to subjectivity. Specific assessment questions have been constructed to assess certain misconceptions, however, other questions may be as adequate in assessing this misconception. The discussion whether a question should be included in the set of question that can possibly detect a misconception is outside of the score of this report. Therefore, we choose to use raw misconception scores rather then normalised scores. Moreover, lack of prior knowledge was not intended to be assessed by the developed test, therefore, no consideration of question construction and efficiency regarding this subject took place.

Table 4.2: Levels of analysis. The levels on which pre- and post test scores have been calculated And per level the defined clusters and the ratings included in calculation of each cluster score. Ratings observed rarely —thus having a minor impact on the cluster scores— are depicted in grey. Ratings that were never provided have been omitted.

Level	Clusters	Response Ratings
Overall	n.a.	all
Class of error	Misconception	H1, H2, H2S3, C2, M1, M2, M2S5, M3S1, S1, H1S1, S2, S3, S4
	Prior knowledge	C, D, E
	Other	-1
Misconception	H2	H2, H2S3
	M1	M1
	M2	M2, M2S5
	M3	M3S1
	S1	S1, H1S1, M3S1
	S2	S2
	S4	S4
Learning goal	**1**	H1, H1S1, H2, H2S3
	2	M1, M2, M2S5, S1, H1S1, M3S1, S2
	3	S3, H2S3
	4	M3S1
	5	C2, S4

4.5 Data analysis

To evaluate the effectiveness of the interactive video in addressing misconceptions about assignment statements for variables in imperative programming held by students in secondary education we compared the mean scores (i.e., the number of incorrect responses) between the pre- and post-test.

Using a Shapiro-Wilk test for normality on the outcome variable a normal distribution could not be assumed for the number of incorrect responses in the post-test, $p = 0.028$, therefore, non-parametric tests were chosen. We used Wilcoxon Signed-ranks tests to compare the means of two related samples.

First of all, we compared the total number of errors between the pre- and post-test to evaluate the overall learning effect. Then, we ran several paired test —comparing the number of errors between the pre- and post-test on various clusters of errors— to further investigate the effectiveness on different levels (see Table 4.2). First, all incorrect responses were grouped by the class of error (i.e., detecting an identified misconception, indicating a lack of prior knowledge, or other mistakes) to investigate the extend to which the learning effect was related to misconceptions. Next, we analysed scores per identified misconception to evaluate the effectiveness in addressing specific misconceptions about variables. Then, we analysed misconceptions scores grouped per learning goal to evaluate the effectiveness in instructing correct understanding of assignment statements for variables.

Lastly, we used a Mann-Whitney U-test comparing separately the pre- or post-test scores between general higher education and pre-university education students on all levels reported in Table 4.2 to investigate whether initial held misconceptions or misconception held after the intervention were dependent on the level of education. Moreover, we ran the (paired) Wilcoxon Signed-ranks test on data split by education level to check whether a possible learning effect was affected by the level of education.

Chapter 5

Results

The interactive video was received overwhelmingly positive by students. They provided comments such as “the video was very educational” or “clear movie!” Multiple students expressed a desire to have received this instruction sooner based on a believe of increased understanding of variables and expectation to be better equipped to complete the programming course assignments.

Identification of misconceptions and development of assessment tests were subject of an earlier report (Plass, 2015), where we also reported the overall effect of our video designed to address identified misconceptions. The difference in number of (in)correct answers between the pre- and post-test indicated that the video positively affected understanding of variables. The majority of participants in the Visual Basic group showed an improvement after the video, however, in the Java and JavaScript samples some participants showed a deterioration in correct knowledge in the post-test. In the remainder of this chapter we report the quantitative results evaluating the effectiveness of the video in addressing misconceptions about assignment statements for variables of primitive type in imperative programming for Dutch secondary education students enrolled in a introductory course Visual Basic programming. We extend evaluation of the effect of the video with detailed analysis for each specific misconception and learning goal.

5.1 Frequencies

A total of 33 participants answers 28 questions each per assessment tests. All but one participants improved their score (the number of correct responses) in the post-test compared to the pre-test. However, there is some variety between participants in the degree of improvement (see Figure 5.1).

In the pre-test non of the participants answers all questions correctly, while in the post-test six participants did so. In total, in the pre-test, participants provided 391 in-

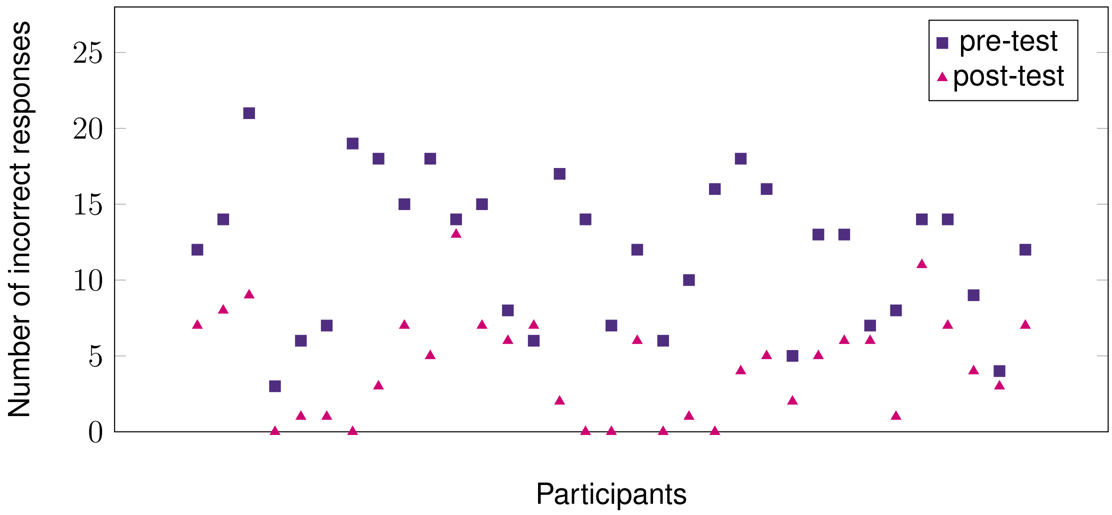


Figure 5.1: Number of incorrect responses (range 0-28) for pre- and post-test per participant.

correct responses, spread over 25 questions. In the post-test, a total of 144 incorrect responses where given over 23 questions, by 27 participants. Only a single question (`n2`) was answered correctly by all participants in both assessment tests. None of the questions was answered incorrectly by all participants in either the pre- and/or post-test, but questions `m1`, `n1` and `n2` never elicited any of the predicted values indicating an identified misconception. In the pre-test 247 incorrect responses, $n = 31$, spread over 20 questions, indicated a identified misconception. In the post-test 59 incorrect responses, $n = 15$, over 18 questions, indicated a identified misconception. The full variation in number of incorrect responses over the different questions, and indicating the identified misconceptions is presented in Table 5.1.

5.2 Data Analysis

5.2.1 Overall Learning Effect

First, we verified earlier results indicating an overall improvement of test results after the intervention. For this we counted and compared the number of incorrect responses per test, using only participants from the Visual Basic course.

A Wilcoxon Signed-ranks test indicated that post-test scores, $Mdn = 5$, were significantly lower than the pre-test scores, $Mdn = 13$, $Z = -5.0, p = 0.00$. In other words, participants showed improved understanding of variables, providing significantly less incorrect responses in the post-test ($Sum = 144$) than the pre-test ($Sum = 391$). This decline in number of incorrect responses aligns with the visual data inspection (Figure 5.1) and indicates a positive overall learning effect, possibly

Table 5.1: Frequency of ratings The number correct responses and the number of incorrect responses indicating identified misconceptions, lack of assumed prior knowledge, or unanticipated behavioural mistakes per question for both the pre- and post-test. | separates between the pre- and post-test ratings. The text colour indicates the differences between the pre- and post-test; black no difference, green improvement and red deterioration. Never observed misconception C1 and S5 are omitted. For non-differential misconception M3 we report the frequencies for M3S1 instead, for all other non-differential incorrect responses we report their occurrence in parentheses under the predominant misconception.

code snippet	question	0 (correct)	-1 (other incorrect)	C Variables have a value retrieved using the name	D Variables are of a certain type	E The default value for a numeric variable is 0	H1 - Variables cannot contain values in conflict with their name	H2 - Variables contain values that were never explicitly assigned	M1 - Variables are set to being equal, also from left to right	M2 - The statement is an equation to be solved	M3S1	C2 - Variables can contain multiple values	S1 - Variables are tested for equality	S2 - The receiving variable is on the right side	S3 - The values of variables are swapped	S4 - The new value is added to the old value
a	-	1|22	1|0	0|0	2|1	22|10	0|0	7|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0
b	-	4|24	2|0	0|0	2|0	14|9	0|0	11|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0
c	-	33|32	0|0	0|0	0|0	0|0	0|1	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0
d	-	32|33	0|0	0|0	0|0	0|0	1|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0
e	1	22|32	3|0	0|0	2|0	0|0	1|0 (H1S1)	1|0 (H2S3)	1|0	0|0	0|0	0|0	3|0	0|1	0|0	0|0
	2	16|31	2|0	1|0	0|0	0|0	1|0 (H1S1)	0|0	1|0	3|0	4|1	0|0	3|0	0|1	0|0	2|0
f	1	27|31	5|1	1|0	0|0	0|0	0|1 (H1S1)	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0
	2	33|31	0|2	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0
g	1	29|32	1|1	2|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	1|0	0|0	0|0	0|0
	2	28|33	0|0	4|0	0|0	1|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0
h	1	20|32	2|0	0|0	0|0	0|0	0|0	0|0	2|0	4|0	0|0	0|0	4|0	1|1	0|0	0|0
	2	12|31	2|0	1|0	0|0	0|0	0|0	0|0	2|0	3|0	6|1	0|0	4|0	1|1	0|0	2|0
i	1	11|31	0|0	0|0	0|0	0|0	0|0	0|0	0|0	4|0	9|1	2|0	4|0	1|1	0|0	2|0
	2	22|32	0|0	0|0	2|0	0|0	0|0	0|0	4|0	1|0	0|0	0|0	3|0	1|1	0|0	0|0
j	1	1|12	1|6	1|2	0|0	2|3	0|0	0|0	0|0	27|10	0|0	0|0	1|0	0|0	0|0	0|0
	2	2|15	3|3	6|4	0|0	2|0	0|0	0|0	0|0	18|8	0|3	0|0	0|0	0|0	0|0	2|0
k	1	3|15	9|4	4|5	0|0	2|1	0|0	0|0	0|0	14|7	1|1	0|0	0|0	0|0	0|0	0|0
	2	2|19	2|2	2|1	0|7	2|3	0|0	0|0	0|0	25|1	0|0	0|0	0|0	0|0	0|0	0|0
l	-	23|30	0|0	1|0	0|0	0|0	0|0	0|0	1|0	1|0 (M2S5)	6|1	0|0	0|0	0|0	0|0	1|2
m	1	29|33	4|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0
	2	32|31	1|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|2	0|0	0|0	0|0
n	1	31|33	0|0	2|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0
	2	33|33	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0	0|0
o	1	18|30	1|1	0|0	1|0	0|0	0|0	0|0	1|0	3|0	6|2	0|0	2|0	0|0	0|0	1|0
	2	25|31	0|0	0|0	3|0	0|0	0|0	0|0	1|0	0|0	0|0	0|0	1|0	0|0	2|2	1|0
	3	11|20	11|11	0|0	0|0	0|0	0|0	0|0	1|0	1|0	6|2	0|0	3|0	0|0	0|0	0|0
p	1	17|28	3|0	0|0	0|0	0|0	0|0	0|0	1|0	0|0	7|3	0|0	3|0	2|2	0|0	0|0
	2	16|23	9|8	0|0	0|0	0|0	0|0	0|0	1|0	2|0	0|0	0|0	3|0	2|2	0|0	0|0

Table 5.2: Frequencies of incorrect responses. The transformed scores, from left to right are presented: the variable and test, the median score, the count of incorrect responses over all participants, and the number of participants ($N = 33$) that provided one or more incorrect responses for the reported score.

Level	**Score *test***	**Mdn**	**Sum**	**n**
Overall	Total *pre-test*	13	391	33
	Total *post-test*	5	144	27
Class of error	Misconception *pre-test*	7	247	31
	Misconception *post-test*	0	59	15
	Prior knowledge *pre-test*	2	82	28
	Prior knowledge *post-test*	1	46	18
	Other *pre-test*	1	62	27
	Other *post-test*	1	39	22
Misconception	H2 *pre-test*	0	19	13
	H2 *post-test*	0	0	0
	M1 *pre-test*	0	16	6
	M1 *post-test*	0	0	0
	M2 *pre-test*	3	106	28
	M2 *post-test*	0	26	12
	M3 *pre-test*	1	45	20
	M3 *post-test*	0	15	5
	S1 *pre-test*	1	82	23
	S1 *post-test*	0	18	6
	S2 *pre-test*	0	8	4
	S2 *post-test*	0	10	4
	S4 *pre-test*	0	11	7
	S4 *post-test*	0	2	2
Learning goal	1 *pre-test*	0	20	14
	1 *post-test*	0	1	1
	2 *pre-test*	6	212	30
	2 *post-test*	0	54	15
	3 *pre-test*	0	2	2
	3 *post-test*	0	2	2
	4 *pre-test*	1	45	20
	4 *post-test*	0	15	5
	5 *pre-test*	0	13	8
	5 *post-test*	0	2	2

caused by the interactive video.

Class of Errors

To analyse further the effect of the intervention on improved understanding of assignment statements for variables of primitive type, we counted and compared the number of incorrect responses classified as either 1) detecting *identified misconceptions*, 2) indicating *lack of assumed prior knowledge*, or 3) *unanticipated behavioural mistakes* as described in Section 4.4.2.

A paired Wilcoxon Signed-ranks test indicated that the post-test scores for identified misconceptions, $Mdn = 0$, were statistically significantly lower than the pre-test scores for identified misconceptions, $Mdn = 7$, $Z = -4.8, p = 0.00$. Moreover, the scores for lack of assumed prior knowledge were significantly lower in the post-test, $Mdn = 2$, than the pre-test, $Mdn = 1$, $Z = -2.2, p = 0.03$. However, there was no significant difference in pre- and post-test scores for unanticipated behavioural mistakes(i.e., unidentified mistakes), $Z = -1.9, p > 0.07$. In other words, participants showed less indication for holding identified misconceptions about variables, and showed increased possession of assumed prior knowledge after watching the interactive video. While there were many incorrect responses corresponding with predicted values mapped to identified misconceptions in the pre-test ($Sum = 247$), there were considerably less in the post-test ($Sum = 59$). A smaller decline in number of incorrect responses was shown for incorrect responses indicating a lack of assumed prior knowledge, and a marginal improvement of unanticipated behavioural mistakes was shown (see also Table 5.2). These results indicate that the video was highly effective in addressing identified misconceptions, but less so for non-anticipated behavioural mistakes.

5.2.2 Effectiveness in Addressing Misconceptions

To investigate the effectiveness of the intervention in addressing specific identified misconceptions we counted the number of incorrect responses corresponding to the predicted value for each misconception, and compared the pre- and post-test scores. Note must be made that all but one misconception were detected for less then 25 participants in the pre-test. Therefore, reported results on the difference between pre- and post-test should not be taken as hard proof of effectiveness but rather should be seen as an indication of learning effect. Misconceptions *H1*, *C1*, *C2*, *S3*, and *S5* have been excluded because they were never or hardly detected in our sample. The scores for the remaining identified misconceptions (i.e., *H2*, *M1*, *M2*, *M3*, *S1*, *S2*, and *S4*) were calculated as described in Section 4.4.2.

A Paired Wilcoxon Signed-ranks test indicated that the post-test misconception scores were significantly lower than the pre-test misconception scores for *H2*, $Z = -3.3, p < 0.01$, *M1*, $Z = -2.3, p < 0.03$, *M2*, $Z = -4.5, p = 0.00$, *M3*, $Z = -3.3, p < 0.01$, and *S1*, $Z = -3.4, p < 0.01$. The post- and pre-test misconception scores did not significantly differ for *S2*, $Z = -0.3, p > 0.76$, and *S4*, $Z = -1.8, p = 0.07$. In other words, five out of seven detected identified misconceptions were diminished after the intervention. Incorrect responses corresponding to predicted values for misconceptions *H2*, *M1*, *M2*, *M3*, and *S1* were less frequently provided in the post-test than the pre-test. While misconceptions *H2* and *M1* were detected in the pre-test, predicted values for these misconceptions were never observed in the post-test (see Table 5.2). Although the predicted values for misconception *M2* were still provided in the post-test, $Mdn = 0, Sum = 26, n = 12$, the frequency strongly decreased compared to the pre-test, $Mdn = 3$—in the pre-test *M2* was overwhelmingly predominantly rated ($Sum = 106$) and detected for the fast majority of participants ($n = 28$). To summarize, misconceptions *H2* and *M1* were fully resolved, *M2*, *M3* and *S1* were significantly positively affected, while *S4* diminished marginally, and *S2* increased minimally after watching the video.

5.2.3 Effectiveness in Instructing Learning Goals

To analyse the effectiveness of the intervention in instructing correct understanding of variables in accordance with the six learning goals as described in Section 3.1 we counted the number of incorrect responses corresponding to the predicted values for all identified misconceptions intended to be addressed by a specific goal, and compared the pre- and post-test scores. Learning goal scores were calculated as described in Section 4.4.2.

A Paired Wilcoxon Signed-ranks test indicated that the post-test learning goal scores were significantly lower than the pre-test learning goal scores for *1*, $Z = -3.3, p < 0.01$, *2*, $Z = -4.5, p = 0.00$, *4*, $Z = -3.3, p < 0.01$, and *5*, $Z = -2.1, p < 0.04$. The post- and pre-test learning goal scores did not significantly differ for *3*, $Z = -0.0, p = 1.00$, and *6*, $Z = -1.0, p > 0.32$. Meaning that, participants provided less incorrect responses corresponding with predicted values for misconceptions intended to be addressed by learning goals *1*, *2*, *4*, and *5* in the post-test than the pre-test. Thus participants showed, after watching the video, better understanding of the meaning of the name of a variable, the structure of an assignment statement, the changeability of a value, and the storage limitations. Student performances did not improve for learning goals *3*, that a value is copied, and *6*, that variables can be part of the expression changing the value of that variable. However, the pre-test scores were already fairly low for learning goals *3*, $Mdn = 0, Sum = 2, n = 2$, and *6*,

$Mdn = 0, Sum = 1, n = 1$. Which can be explained by the fact that these learning goals have been defined to address respectively *C1* and *S3*, and *S5*, which were hardly or never rated.

5.2.4 Interaction Effects

Level of Education

We investigated the possible moderating effect of level of education by comparing pre- and post-test scores separating participants in higher general education (HAVO, $n = 22$) from those in pre-university education (VWO, $n = 11$). By splitting into two sub-samples the samples sizes are lower then 25, therefore, results should not be read as hard proof but rather give an indication.

First, a Mann-Whitney U-test was conducted to compare the overall pre-test scores for HAVO and VWO students. There was no significant difference in the overall pre-test scores for HAVO ($Mdn = 12.5$) and VWO ($Mdn = 13$) students, $Z = -0.61, p > 0.54$. Then, a Mann-Whitney U-test was conducted to compare the overall post-test scores for HAVO and VWO students. Once more, there was no significant difference in the overall scores for HAVO ($Mdn = 3.5$) and VWO ($Mdn = 6$) students, $Z = -0.92, p > 0.35$. These results suggest that, initially, HAVO and VWO students hold similar (mis)understandings of variables in imperative programming, and that the intervention was as effective for HAVO as for VWO students.

More detailed analyses revealed no differences between HAVO and VWO students in misconception scores, individual identified misconceptions, learning goal scores, or unanticipated behavioural mistakes for both pre- and post-test. With exception of learning goal *3*, for which there was reported a significant difference between HAVO ($Mdn = 0, M = 0.0$) and VWO ($Mdn = 0, M = 0.18$) students, $Z = -2.03, p < 0.05$. However, learning goal *3* scores were calculated from *C1* and *S3* scores who have been rated respectively zero and four times in total. Although, the samples are to small to base any conclusion, these results strengthen the previous suggestion that the effectiveness of the intervention in addressing identified misconceptions was not dependent of or influenced by the level of education.

Noteworthy, Mann-Whitney U-tests did show significant differences in pre-test assumed prior knowledge scores between HAVO ($Mdn = 1, M = 1.82$) and VWO ($Mdn = 3, M = 3.82$), $Z = -2.66, p < 0.01$, and in post-test assumed prior knowledge scores between HAVO ($Mdn = 0, M = 0.77$) and VWO ($Mdn = 3, M = 2.64$), $Z = -2.86, p < 0.01$. In other words, VWO students made significantly more mistakes indicating a lack of assumed prior knowledge then HAVO students in both the pre- and post-test. This difference between HAVO and VWO students is also reflected in Wilcoxon Signed-ranks test conducted on the split sample; indicating that in the HAVO sample the post-test assumed prior knowledge scores were significantly lower than the pre-test assumed prior knowledge scores, $Z = -2.18, p < 0.03$, while in the VWO sample the assumed prior knowledge scores did not significantly differ between the post- and pre-test, $Z = -0.93, p > 0.35$. Meaning that the HAVO students initially showed less indications of a lack of assumed prior knowledge, and further improved their assumed prior knowledge after the intervention, than VWO students.

Chapter 6

Discussion

The results strongly indicate that the interactive video was effective in addressing identified misconceptions about assignment statements in variables of primitive type for secondary education students enrolled in a introductory programming course.

Although evaluation of the assessment tests was subject to an earlier report and outside the scope of this document, we wish to emphasise that there were some limitations towards these tests abilities to detect misconceptions. First, not all identified misconceptions were detectable by a equal number of questions; the most extreme case, misconception $\mathrm{S5}$, was uniquely detectable by only one question. Moreover, incorrect responses were not seldomly observed in different questions than those designed to assess a certain misconception (see Table 5.1). Secondly, many predicted values mapped to certain identified misconceptions did not differentiate between two or more misconceptions, and sometimes between the correct answer (see Table 2.2). Lastly, a relatively large number of observed incorrect responses were not mapped to identified misconceptions; in the post-test over half of the mistakes indicated a lack of assumed prior knowledge or could not be identified (see Table 5.2). Due to these limitations, hindering detection of misconceptions held in the sample, it seems likely that not all misconceptions held by students have been effectively detected — both before and after watching the interactive video. Further, not all identified misconceptions may have been held by the students in our sample. The container analogy was not instructed to these students (who all reported no prior programming experience) and thus less likely to be hold. Therefore, never detected misconception $\mathrm{C1}$ and misconception $\mathrm{S5}$ have been omitted from this discussion, as is **learning goal 6** which was planned being evaluated by misconception $\mathrm{S5}$ score only. In the remainder of this chapter we interpret the results for the remaining identified misconceptions and learning goals in the light of these limitations.

6.1 Learning Effect of Interactive Video

Overall, students showed improved understanding of variables, providing less incorrect responses mapped to identified misconceptions or indicate lack of assumed prior knowledge after watching the video compared to before. Before an in depth discussion on the effectiveness of the intervention on specific learning goals and identified misconceptions, we first discuss some of the noteworthy findings for individual students and assessment questions.

Class of Errors In total students provided less incorrect responses in the post-test ($Sum = 144$) compared to the pre-test ($Sum = 391$). Of these, in the pre-test 247, and in the post-test 59, incorrect responses mapped to the predicted values indicating an identified misconception. This means that while the overall the number of incorrect responses decreased by 63%, those mapped to identified misconceptions even decreased by 76%. This indicates that the video was highly effective in addressing identified misconceptions. However, the video was less effective in helping students to overcome unanticipated behavioural mistakes. In the pre-test a total of 27 participants made 62 unanticipated mistakes, compared to 39 mistakes by 22 students in the post-test. With a 37% decrease students still showed to improve on their unanticipated behavioural mistakes, but lesser than for those mapped to identified misconceptions. Also for incorrect responses related to assumed prior knowledge, with a nearly 45% decrease, the video showed helpful but not as effective as for identified misconceptions. This is not surprisingly considering the fact that the video was designed to instruct learning goals explicitly addressing identified misconceptions, less so teaching assumed prior knowledge and not addressing unidentified misconceptions underlying unanticipated behavioural mistakes.

Individual Students All but one participant performed better on the post-test than the pre-test. Participant `V0015JA` provided six incorrect responses (mapping 2 to prior knowledge E and 4 to misconception $\mathrm{M2}$) in the pre-test and seven (mapping 3 to prior knowledge E, 3 to misconception $\mathrm{M2}$ and 1 to misconception $\mathrm{S4}$) in the post-test. These small numbers do not inform any suggestions on the relation between the intervention and deteriorating performance; the student might as well have been tired or distracted. Further, while in the pre-test none of the participants provide all correct responses, in the post-test six participants did so. Additionally, 13 students provided five or less incorrect responses in the post-test. Given that some of these students provided as much as 19 incorrect responses in the pre-test we can safely assume that students improved their individual performance during the intervention.

Individual Questions In total, 24 out of 28 questions were answered correctly more frequently in the post-test than the pre-test. One question, `n2`, was answered correctly by all students in both the pre- and post-test. The remaining three questions received one or two more incorrect responses in the post-test than the pre-test. Question `c` was answered incorrectly in the post-test by one participant, `H004BB`, who provided the predicted value for misconception $\mathrm{H1}$. Also in the pre-test, this participant provided once the predicted value for misconception $\mathrm{H1}$, but then for question `d`. Question `f2` was answered incorrectly by none of the participants in the pre-test and by two in the post-test; both incorrect responses could not be identified as detecting an identified misconception or lack of assumed prior knowledge. Question `m2` received one (unidentified) incorrect response in the pre-test, while in the post-test two participants provided the predicted value mapped to misconception $\mathrm{S1}$. Interestingly, for both question `post_f2` and `post_m2`, one of the two participants who provided an incorrect response was participant `H004BB`. This participant provided 0 for all three questions. Alternatively, to showing to hold misconception $\mathrm{H1}$ form question `post_c`, or misconception $\mathrm{S1}$ form question `post_m2`, s/he may have chosen the Visual Basic default value for Integers whenever unable to give the correct value. The fact that variables have a default value, assumed prior knowledge prior knowledge E, appeared not fully understood by the participant in the pre-test, while in the post-test s/he did not provide any incorrect responses indicating a lack of assumed prior knowledge.

6.1.1 Instructing Correct Understanding

In this section we investigate the results per learning goal constructed to address specific identified misconceptions and leading the scenario of the interactive video.

1. The name of a variable is of no meaning. Next to a significant improvement in student performance, also in the post-test there was only a single incorrect response that mapped to the predicted value indicating a misconception addressed by **learning goal 1**. However, upon further inspection of the responses for related questions `a`–`f` for this participant, `H004BB`, (see Appendix D Section D.1), its seems unlikely that the participant's incorrect response was caused by the name of the variable. Rather it seems s/he is using `0` as a default responses whenever not knowing the correct answer. Therefore, we believe that all students had appropriate knowledge of **learning goal 1** after watching the video which indicate that the video was effective in instructing **learning goal 1**.

2. A value is assigned to a variable, using the name of the variable followed by the $=$ symbol and the expression for the value. Also for **learning goal 2** the video is believed to be highly effective. Even though student performance did not improve for all related identified misconceptions (see Section 6.1.2), improvement on remaining identified misconceptions was strong enough to result in an overall improvement for **learning goal 2**. While in the pre-test the majority of participants ($n = 30$) provided incorrect responses indicating holding one or more identified misconceptions hindering correct understanding of value assignment, this diminished to under half of all participants for the post-test ($n = 15$). Further, participants, on average, made significantly less incorrect responses mapped to the predicted value for related identified misconceptions in the post-test ($Mnd = 0$) than the pre-test ($Mnd = 6$). Although participants still showed to have not fully grasped the assignment statements, both the number of participants showing problems and the severity of misunderstanding decreased after watching the interactive video. This indicates that the interactive video contributed to improved understanding of **learning goal 2**.

3. A value is copied to a variable. The predicted values detecting identified misconceptions related to **learning goal 3** were hardly ever observed in our sample. Either the tests were ineffective assessing these misconceptions, or the misconceptions were not held by the students. As a result, no difference was found between the pre- and post-test scores for **learning goal 3**. From this follows, we cannot conclude about the effectiveness of the interactive video in instructing **learning goal 3**.

4. The value of a variable can be changed at runtime. While in the pre-test over 60% of the participants ($n = 20$) provided one or more incorrect response(s) that matched the predicted value for the identified misconceptionaddressed by **learning goal 4**, in the post-test only a few participants ($n = 5$) remained to show signs of incorrect understanding of **learning goal 4**. Moreover, while in the pre-test six participants provided more than one incorrect response , only the two participants, `H0012JR` and `H0032JH`, remained to provide a relatively large number of incorrect responses mapped to the misconception related to **learning goal 4** (resp. 6 and 7). Based on the significant overall improvement, and considering absence of improvement for few individual students, we may conclude that the interactive video was effective in instructing **learning goal 4** to most students, however, it may be less effective for students holding firm misconceptions hindering correct understanding. However, **learning goal 4** addressed only misconception $\mathrm{M3}$, which could never be uniquely detected by the assessments tests. The given incorrect responses could alternatively indicate misconception $\mathrm{S1}$. Only two of the six participants providing more than once the predicted value for either misconception $\mathrm{M3}$ or misconception $\mathrm{S1}$, provided one or two times an incorrect responseuniquely detecting misconception $\mathrm{S1}$; this is not enough to draw any conclusion on the likely hood of this student holding either misconception $\mathrm{M3}$ or misconception $\mathrm{S1}$ or both. Therefore, in order to provide sound conclusions on the effectiveness of the video in instructing **learning goal 4**, improved assessment tests able to uniquely detect whether a students hold misconception $\mathrm{M3}$ are required.

5. A variable can store one (1) value, on change the previous value is lost. From the pre-test there were no strong indications of students lacking understanding of **learning goal 5**, or holding misconceptions hindering correct understanding thereof. With a fairly low number of 15 responses mapping the predicted values for related identified misconceptions, provided by a total of 8 participants and scattered over six question we have not enough data to suggest students having trouble under standing **learning goal 5**. Surprisingly, considering indications for holding misconception $\mathrm{C2}$ were hardly observed, and there was no significant effect reported for misconception $\mathrm{S4}$, nonetheless, pre- and post-test scores for **learning goal 5** were significantly different. Apparently, the combined effect of misconception $\mathrm{C2}$ and misconception $\mathrm{S4}$ is strong enough, however, we consider results based on such small numbers sketchy at least. Even though, the results show a small but significant improvement for **learning goal 5** in the post-test compared to the pre-test, based on the small numbers, we believe more research is needed to evaluate the effectiveness of the video in instructing **learning goal 5**. Preferably with a sample holding misconception $\mathrm{C2}$, which is one of the two addressed identified misconceptions, but was never detected in our sample.

6. The value of a variable can be changed as the result of an expression using the value stored in this variable. This learning goalcannot be evaluated because related misconception $\mathrm{S5}$ was not assessed adequately by the tests.

To summarise, before watching the video, one or more misconceptions hindering correct understanding of **learning goal 2** and **learning goal 4** were held by the majority of students. Additionally, some indications for having problems understanding **learning goal 1** and **learning goal 5** were held by some students. Overall, the scores for these learning goals were improved after watching the video. Although for **learning goal 1**, **learning goal 2**, and **learning goal 5** one or more identified misconceptions were not held or improved in our sample, the combined remaining misconception(s) appeared strong enough to affect their related learning goalresult. After the video students show significant improved understanding for **learning**

goal 1, that the name is of no meaning, **learning goal 2**, that a value is assigned from an expression right from the $=$ symbol, and **learning goal 4**, that the value can change at runtime. This indicates that these learning objectives were sufficiently instructed in the interactive video. No significant effects were reported for **learning goal 3**, that a value is copied, and **learning goal 6**, that the variable may be mentioned in the expression. This is easily explained by the fact that no indications for holding the misconceptions addressed by these learning goals were present in our sample. If a student holds correct understanding to begin with, a video cannot increase understanding, and differences in pre- and post-test score were not expected. A student sample holding related misconceptions would be needed to evaluate the effect of the video for these learning goals.

6.1.2 Addressing Misconceptions

In this section we investigate to what extend the intervention addressed identified misconceptions by discussing students's performance for each misconception.

Detected Misconceptions Initially some identified misconceptions appeared more prevalent in our sample than others. Especially, misconception $\mathrm{M2}$, that a statement is an equation to solve, received a high score, meaning that misconception $\mathrm{M2}$ was held strongly and by the majority of students. This was in line with our expectations; having observed students struggle with math analogies while writing code assigning a value resulting from some computation. Although, from the same observations, we would expected a high score for misconception $\mathrm{S5}$ as well, this was not the case because misconception $\mathrm{S5}$ could never be uniquely detected from the participant responses in the assessment test. Further, misconception $\mathrm{M3}$ and/or misconception $\mathrm{S1}$, were detected relatively frequent troughout the pre-test responses. Sadly, the tests lacked the ability to understand whether the students holds misconception $\mathrm{M3}$ or misconception $\mathrm{S1}$ because the predicted value did never differentiate between the two. In some occasions, however, misconception $\mathrm{S1}$ was uniquely detected by unanticipated alternative incorrect responses. On the contrary, misconception $\mathrm{H1}$, misconception $\mathrm{C1}$, and misconception $\mathrm{C2}$, misconception $\mathrm{M1}$, misconception $\mathrm{S2}$, misconception $\mathrm{S3}$, and misconception $\mathrm{S4}$ all received low scores, meaning that predicted values were never, hardly, or irregularity observed. This may mean that either students did not hold these misconceptions or the assessment tests could not detect these misconceptions sufficiently. The fact that misconception $\mathrm{C1}$ and misconception $\mathrm{C2}$ were hardly observed can be explained by the fact that the container analogy was not instructed to these students and all students reported to have no prior programming experience. Although, the over all score for misconception $\mathrm{H2}$ was relatively low, all rating were given in questions `a` and `b`, the questions designed to assess whether a student holds this misconception. About a third of the participants provided the predicted value mapped to misconception $\mathrm{H2}$ for each of these questions, indicating that at least some students hold this misconception.

$\mathrm{H1}$. Variables cannot contain values in conflict with their name. This misconception was rated only in few times in both the pre- and post-test. Based on our data we cannot make any conclusions on the effectiveness of the video; from the pre-test we did not get enough evidence that students were holding this misconception, and the intervention cannot solve what is not there.

$\mathrm{H2}$. Variables contain values that make sense given their name, but were never explicitly assigned. While 13 students showed some signs of holding misconception $\mathrm{H2}$ by providing predicted values mapped to misconception $\mathrm{H2}$ for questions `pre_a` and `pre_b`, none did so in post-test. After watching the video, none of the participants though the variable 'magically' received a value fitting its name. From this we may conclude that the intervention effectively addressed misconception $\mathrm{H2}$.

$\mathrm{M1}$. Variables are set to being equal, also from left to right. The results show a significant improvement in scores for misconception $\mathrm{M1}$, however, when we investigate the individual student responses there is some thing interesting to see. The majority of the predicted values mapped to misconception $\mathrm{M1}$ are given by one student; participant `H0024WW` is responsible for 10 of the total 14 responses indicating misconception $\mathrm{M1}$. This means that this participant had a strong believe that values could also be assigned from left to right. Given that s/he never provided s predicted value indicating holding misconception $\mathrm{M1}$ in the pre-test we can conclude that this participant's believed were successfully transformed. Moreover, none of the participants provided any incorrect responses indicating holding misconception $\mathrm{M1}$ in the post-test. This indicates that the intervention was effective in addressing misconception $\mathrm{M1}$.

$\mathrm{M2}$. The statement is an equation to be solved. The majority of participants seemed to be tricked by the equation-like appearance of an assignment statement. Only five out of 33 participants did not provide the predicted values mapped to the questions designed to assess this misconception (i.e., `pre_j1`, `pre_j2`, `pre_k1`, `pre_k2`). However, in the post-test 21 participants did not provide the predicted values indicating misconception $\mathrm{M2}$. This means that 16 participants who did show indications for holding misconception $\mathrm{M2}$ no longer did so in the post-test. This, and the significant results presented in Section 5.2.2, led us to believe that the video was effective in addressing misconception $\mathrm{M2}$. However, after watching the video still a relatively high number of incorrect responses matching the predicted values for misconception $\mathrm{M2}$ was observed for post-test questions `j1`–`k2` (i.e., 26 out of 132 possible responses). Noteworthy, in the post-test none of the participants provided the predicted values mapped to misconception $\mathrm{M2}$ for other questions that those designed to assess this misconception. Given the number of incorrect responses, these questions where the hardest in the tests; of 132 responses, in the pre-test only eight were correct, in the post-test 61. Nonetheless, the number of incorrect responses matching the predicted values for misconception $\mathrm{M2}$ was lower in the post-test ($n = 26$) than the pre-test ($n = 84$). However, the number of incorrect responses not mapped to identified misconceptions slightly increased. This may indicate that the video was effective in addressing misconception $\mathrm{M2}$ but did not provide students with appropriate correct models to allow them to answer the questions correctly. Even though, misconception $\mathrm{M2}$ was not fully resolved, and students seemed to lack knowledge to apply alternative correct strategies, there was still a significant deterioration in students holding this misconception. Therefore, we believe the video to be a good first step but the remaining incorrect responses responses from the post-test should be investigated in order to see what problems students hold. Additional material of adaption of the interactive video should be considered to address these 'new' misunderstandings.

$\mathrm{M3}$. Variables are fixed values or constants, assigned a value once. The results show a significant positive effect for misconception $\mathrm{M3}$. However, because the predicted values for misconception $\mathrm{M3}$ and *S1* were non-differential we cannot be sure which misconceptionstudents were holding. Therefore, we also cannot be sure whether the intervention had any effect on students holding misconception $\mathrm{M3}$.

$\mathrm{C1}$. A value is moved, a variable on the right side loses the value it contained. This misconception was never rated in both assessment tests. Therefore, we cannot say anything about the effectiveness of the intervention on addressing misconception $\mathrm{C1}$.

$\mathrm{C2}$. Variables can contain multiple values, like a box can contain multiple items. This misconception was rated twice in the pre-test and never in the post-test. Based on this data we cannot conclude anything about the effectiveness of the intervention on addressing misconception $\mathrm{C2}$.

$\mathrm{S1}$. Values are tested for being equal, which is true or false. The results show a significant positive effect after watching the video for misconception $\mathrm{S1}$. While in the pre-test the majority of participants ($n = 23$) provided one or more incorrect responses that might indicate misconception $\mathrm{S1}$, only six participants did so in the post-test. Moreover, the total number of incorrect responses indicating misconception $\mathrm{S1}$ declined from 82 in the pre-test to 18 in the post-test. However, the score for misconception $\mathrm{S1}$ was heavily based on predicted values that do not differentiate between misconception $\mathrm{S1}$ and misconception $\mathrm{M3}$. In the pre-test 47 out of 82 incorrect responses matched the predicted values for misconception $\mathrm{S1}$ and misconception $\mathrm{M3}$, 35 incorrect responseincluded an unanticipated response that did uniquely detect misconception $\mathrm{S1}$. These incorrect responses, `Not equal Error`, as were frequently observed in the pre-test were never observed in the post-test. This gives strong indication that after watching the video students no longer mistakenly thought of assignment statements as equations to solve. Thus we believe the intervention to have effectively addressed misconception $\mathrm{S1}$.

$\mathrm{S2}$. The receiving variable is on the right side. This misconception was the only one showing a minor insignificant deterioration. Noteworthy, none of the four students who provided a total of eight responses indicating misconception $\mathrm{S2}$ in the pre-test was amongst the four students who provided 10 responses indicating misconception $\mathrm{S2}$ in the post-test. This may suggest that the video was effective in addressing this misconception to students already holding misconception $\mathrm{S2}$, however, other students may have developed misconception $\mathrm{S2}$. Further research with a sample initially clearly holding misconception $\mathrm{S2}$ is needed to form any conclusions about the effectiveness of the video in addressing this misconception.

$\mathrm{S3}$. The values of the variables are swapped. This misconception was rated hardly in both the pre- and post-test. Based on this data we cannot conclude anything about the effectiveness of the intervention on addressing misconception $\mathrm{S3}$.

$\mathrm{S4}$. The new value is added to the previous value. Although there was a slight decrease in incorrect responses that matched the predicted values mapped to misconception $\mathrm{S4}$ in the post-test ($n = 2$) compared to the pre-test ($n = 11$), the rating are so infrequent that we cannot draw any conclusions about the effectiveness of the interactive video in addressing misconception $\mathrm{S4}$.

$\mathrm{S5}$. Results can only be stored in variables not mentioned in the expression on the right side. This misconception was never uniquely identifiable in both assessment tests. Therefore, we cannot say anything about the effectiveness of the intervention on addressing misconception $\mathrm{S5}$.

To summarise, for all but one misconception participants showed improvement, however, not all misconceptions were initially detected making it impossible to investigate the effect of the video on these misconceptions (see Table 6.1). Participants in our sample seem to initially mostly hold math-related misconceptions, on which all three showed significant lower scores in the post-test meaning that less incorrect responses indicating these misconceptions where provided. Moreover, improvements were found for misconception $\mathrm{H2}$ and misconception $\mathrm{S1}$. Misconceptions $\mathrm{H1}$, $\mathrm{C1}$, $\mathrm{C2}$, $\mathrm{S2}$, $\mathrm{S3}$, $\mathrm{S4}$, and $\mathrm{S5}$, were hardly or never rated in our sample. These findings strengthen our believe that the video was effective in addressing misconceptions, if a misconception was clearly held by multiple students before, it was diminished after watching the video.

6.1.3 Interaction Effects

Our participant sample was not homogeneous in gender and education level which may skew results because they are influenced by certain characteristics of the student. Moreover, the majority of participants provided some incorrect responses that may indicate a lack of assumed prior knowledge. Ideally, one would test directly the relation between education level and learning effect and lack of assumed prior knowledge , however, with the current data set an appropriate test could not be found. Instead, results for the influence of education level have been derived from separate analysis investigating pre- and post-test scores for pre-university students and for general higher education students.

Education Level was not found to have affected the learning effect from the intervention on addressing misconceptions, nor influence initially held misconceptions. If any, somewhat surprisingly, overall pre-university students performed, although insignificantly, worse than general higher education students.This phenomena cannot be explained by subjective perception of the students about the video; both general higher education students and pre-university students provided positive comments about the video. During the programming course, pre-university students did have less classroom hours and thus less programming experience and examples. This may explain why knowledge about variables in programming was lower for pre-university students than general higher education students. One might argue that pre-university students have more advanced math knowledge, making it more likely for them to use math models where programming models are lacking. However, inspection of individual students's results provide no evidence for this hypothesis; pre-university students incorrect responses are not necessarily more frequently rated misconception $\mathrm{M1}$, misconception $\mathrm{M2}$, and misconception $\mathrm{M3}$.

Table 6.1: Updated overview of the identified misconceptions and related learning goals addressing these misconceptions, indicating degree of overall performance increase by students in the post-test compared to the pre-test. Items with a significant improvement are coloured green; darker green indicated 0 errors in post-test. Items with an insignificant improvement are coloured orange. Items with an insignificant deterioration are coloured red. Grey items were hardly or never observed in the tests results.

Learning goals	Mathematics	Human interaction	Container analogy	Semantics
1. The name of a variable is of no meaning.		H1. Variables cannot contain values in conflict with their name. H2. Variables contain values that make sense given their name, but were never explicitly assigned.		
2. A value is assigned to a variable, using the name of the variable followed by the = symbol and the expression for the value.	M1. Variables are set to being equal, also from left to right. M2. The statement is an equation to be solved.			S1. Values are tested for being equal, which is true or false. S2. The receiving variable is on the right side.
3. A value is copied to a variable.			C1. A value is moved, a variable on the right side loses the value it contained.	S3. The values of the variables are swapped.
4. The value of a variable can be changed at runtime.	M3. Variables are fixed values or constants, assigned a value once.			
5. A variable can store one (1) value, on change the previous value is lost.			C2. Variables can contain multiple values, like a box can contain multiple items.	S4. The new value is added to the previous value.
6. The value of a variable can be changed as the result of an expression using the value stored in this variable.				S5. Results can only be stored in variables not mentioned in the expression on the right side.

Lack of Prior knowledge appeared higher for pre-university students than general higher education students. According to the constructivist approach (Ben-Ari, 1998; Kuittinen & Sajaniemi, 2004) a lack of prerequisite knowledge hinders construction of viable models for new knowledge. This suggests that students who lack assumed prior knowledge are less capable of learning more advanced topics, thus the intervention should be less effective for pre-university students compared to general higher education students. Although pre-university students did show significantly more lack of assumed prior knowledge, and did not improve upon assumed prior knowledge after watching the video, this did not affect their ability to advance knowledge explicitly instructed in the video. Pre-university students did not differ from general higher education students in misconceptions held before and after watching the video. This indicates that the interactive video is effective in instructing viable models, addressing misconceptions, independent of the status of assumed prior knowledge. However, to properly test this hypothesis, it is required to compare between participants who do lack assumed prior knowledge and those who don't. Measurement of assumed prior knowledge was not part of the study design, therefore, we don't have a clear metric defining the two groups. Moreover, the majority of participants provided some incorrect responses that may indicate lack of assumed prior knowledge . As a result, we would end up with almost no participants in the hypothetical 'no lack of assumed prior knowledge' group. Ideally, manipulation of these groups would take place by running the study in two groups where only one receives proper instruction of assumed prior knowledge before the experiment. Further, by design, at the start of the video the assumed prior knowledge was briefly introduced, this minimal information may have been enough to trigger existing knowledge and thereby resolve any appearing gaps in assumed prior knowledge. This was a conscious decision to maximise the possibility of effectively addressing misconception. From the fact that, in general, lack of assumed prior knowledge diminished, we may conclude that this strategy was effective. However, if the video resolved lack of assumed prior knowledge at its start, it can no longer negatively influence the learning effect.

6.2 Limitations

Our study proposes some limitations with respect to the measurements and sampling that may have affected the validity.

Measurement First, evaluation of the interactive video is highly dependent on quantitative data obtained from the assessment tests. Qualitative analysis was limited to a single optional question about students' perception of the video. Proper triangulation could have strengthen the validity of the research. Second, the assessment tests were incapable of detecting each identified misconception to the same degree. Different misconceptions could possibly be detected by a different number of question; misconception $\mathrm{S5}$ could be detected only by `1`, whereas, for example, misconception $\mathrm{S4}$ was possibly detectable by ten questions. As a result some misconceptions were more likely to be detected than others, and the frequency score (i.e., the number of times a misconception was rated) was not necessarily representative for the strength to which a misconception is held. Alternatively, we could have calculated frequency scoresbased on only the questions designed to assess a certain misconception. However, this would mean rejecting valid responses and decreasing the statistical power. For example, misconception $\mathrm{S1}$ was rated frequently in the pre-test ($Sum = 37$), but never on the intended questions (i.e., pre_f1, pre_f2, pre_m1, pre_m2). Further, the incorrect responses did not always differentiate between misconceptions, hindering unique detection of specific misconceptions. For example, the predicted values mapped to misconception $\mathrm{M3}$ were always the same as for misconception $\mathrm{S1}$. These limitations could have been prevented by a pilot test evaluating the assessment tests. Which could have improved the validity of misconception assessment before and after watching the video. Lastly, the rating from participants' responses to misconceptions was done by only two experts. Although rating was done based on a formalised schema, and visual inspection showed a high degree of inter-annotator agreement, there remains some degree of subjective interpretation. For example, the response `0.5` to `pre_e1` was not included in the schema and could be interpreted as holding misconception $\mathrm{M2}$ or a sign of lack of prior knowledge D. Different annotations affect the scores, as for the example, potentially misconception $\mathrm{M2}$ is underestimated. Rating validity can be optimised by an increased number of annotators rating the same responses and then formally test for inter-annotator agreement.

Sampling The participant sample was relatively small ($N = 33$), non-homogeneous, and non-representative for the target population (i.e., secondary education students in the Netherlands). All participants were recruited from the same school and received the same lessons from the same teacher prior to participation. This affected the initially held misconceptions, for example, the container analogy was not instructed which likely contributed to the low frequency scores for misconception $\mathrm{C1}$ and misconception $\mathrm{C2}$. Repeated studies in different schools are needed to evaluate the effectiveness of the video in addressing identified misconceptions not detected in the current sample. Further, there was an unequal distribution in gender and educational level within the sample. No indications were present that these participant characteristics influenced the effects. However, the sample sizes were to small to have enough power for valid statistical analysis. The statistical power can easily be increased by raising the number of participants.

Control Condition Due to the fairly low number of participants and inclusion of the study as part of the course, our study did not include a control condition. As a result, it remains unclear if the observed learning effect results from watching the interactive video or participation in the study as a whole. Students may learn from making the assessment tests , or the novelty can negatively affect pre-test scores (i.e., unfamiliarity with the question format and nervousness from the new experience may result in more incorrect responses). The post-test questions followed the same structure as the pre-test but used different variable names and values to minimise the learning effect. Nonetheless, advancing over the code snippets, the student might suddenly 'get it', as was reflected in comments given after the test such as "I made a mistake at question 17, the answer should be `0`". The limitations presented above can be overcome by adding warm-up questions that are not part of the study data, and inclusion of a control group making both assessment tests, without receiving the intervention in between. In a similar vein, the effectiveness of the interactive video could be compared with lessons on variables from existing materials by including another group that receives this lesson in between the two tests. Moreover, a retention-test to evaluate the effectiveness over time would be worthwhile.

Chapter 7

Conclusion and Recommendations

7.1 Conclusions

Assignment statements for variables of primitive types is one of the basic concepts in imperative programming, but this concept causes a lot of problems for secondary education students in the Netherlands. In an empirical study we found that a relatively short (7 min.) interactive video step-by-step instructing assignment statements effectively served as an *antidote* for misconceptions that students have from their experiences in other domains. Also the video was verbally appreciated by students.

Based on literature we identified four classes of misconceptions students may hold and can hinder correct understanding: mathematics, anthropomorphism, container analogy, and semantics. To detect misconceptions held by individual learners we developed assessment tests which consisted of 16 programming code snippets with variable declarations and assignment statements and 28 short-answer questions. To transform misconceptions into viable models we developed an interactive video instructing six learning goals trough out the unfolding of Nellie's attempt finding Nico's password. The video was made in a paper-cut style and illustrates the values of variables in the computer memory whilst executing statements. The video was paused on three key-moments, offering a multiple-choice question to focus attention and optimise knowledge gain. The effectiveness of this video in addressing identified misconceptions was evaluated with 33 students of general higher or pre-university education at the Candea College in the Netherlands.

We found a positive effect of the intervention on students understanding of variables; identified misconceptions detected prior to watching the video were observed significantly less afterwards. Initially misconceptions related to mathematics were predominantly held by the students, after watching the video all three math-related misconceptions were significantly less observed. Of the two anthropomorphism-related misconceptions, only misconception $\mathrm{H2}$ was initially detected, and it was fully resolved after watching the video. The semantics-related misconceptions were detected inconsistently, and misconceptions from the container analogy were hardly ever detected. Further research is needed to evaluate the effectiveness on identified misconceptions not detected in our sample or not assessed by the tests. Moreover, follow-up research should include a control group to verify that the positive learning effect is caused by the video and not by other parts of the experiment procedure.

The effectiveness of the video on addressing identified misconceptions was not influenced by gender, educational level or prior knowledge. Contrary to suggestions from literature (e.g., LEM effect (Robins, 2010), ZPD (Vygotsky, 1980)), all students improved their score, independent of initially lacking assumed prior knowledge. The video even showed to improve students's assumed prior knowledge, but unanticipated behavioural mistakes were not influenced by the intervention. Noteworthy, lack of prior knowledgeshowed some relation with level of education, surprisingly, it was initially observed more in pre-university students than general higher education students, and the intervention was less effective in resolving lack of assumed prior knowledge for pre-university students than general higher education students. Further research is needed to verify if this is a trait of pre-university education level or caused by other coexisting variables such as the amount of programming experience prior to the study.

To conclude, a well designed video instruction can significantly improve students's knowledge about various aspects of a programming concept such as variables. Investigating possible misunderstandings, defining learning goals to address these misconceptions, and gradually instructing this knowledge supported with visual aids to teach not only correct syntax but also conceptual knowledge was proven a success full approach to teach assignment statements for variables in imperative programming to secondary education students.

7.2 Recommendations

The present study and developed materials have shown to be valuable, nonetheless, with some minor improvements the validity of the study can be increased, as are there challenges remaining for the future.

Improvements First of all, inclusion of control condition is advised to validate whether the observer learning effect was caused by the interactive video. Secondly, repeating the study at various schools allows for analysis of influence from teacher and course material on initially held misconceptions and learning effect. Having a larger sample also improves the accuracy of analysis for other mediator variables such as level of education. Third, expanding the qualitative measurements to an obligatory survey provides insight into the students perception of the interactive video, this can strengthen the confidence of the evaluation result. Additionally, a retention test could be included to evaluate the sustainability of the learning effect. Lastly, the assessment tests require revision and should be evaluated in a pilot test prior to evaluation of the educational material. Although suggested improvements to the assessment tests have been given in an earlier report (Plass, 2015, pp. 44–45), we wish to emphasize that adjustment of the programming code snippets and questions so that all identified misconceptions are detectable by multiple questions and that the predicted values differentiate between specific identified misconceptions is necessary.

Future Work Based on the indicated positive learning effect of the interactive video a next step would be to embed the video in the course design. This requires to pinpoint the optimal moment to teach the learning goals instructed in the present video. This optimal moment may be programming language and/or course material dependent. Further, possibly other type of variables and programming concepts can be instructed by videos likewise. Ultimately, a new imperative programming course method could be designed around various videos instructing individual concepts. Embedding and or extending upon the interactive video will raise research questions such as the following.

a To what extend does the video facilitate knowledge transfer; does course work performance benefit from learning effect?

b To what extend is the learning effect influenced by the timing of presentation of the video within the course program?

c How effective is an interactive video instructing other types of variables?

d How effective is an interactive video instructing other programming concepts?

e To what extend are videos perceived as interesting by the students when offered more frequently (for various concepts throughout the course)?

References

Bayman, P., & Mayer, R. E. (1983). A diagnosis of beginning programmers' misconceptions of BASIC programming statements. *Communications of the ACM*, *26*(9), 677–679.

Ben-Ari, M. (1998). Constructivism in Computer Science Education. In *Sigsce'98* (pp. 257–261). Atlanta: ACM.

Bornat, R. (2014). *Camels and humps: a retraction *.*

Dehnadi, S. (2006). Testing programming aptitude. In *Proceedings of the 18th annual workshop of the psychology of programming interest group* (pp. 22–37).

Dehnadi, S., & Bornat, R. (2006). *The camel has two humps.*

Ferguson, C. (2014). *The camel doesn't have two humps: Programming aptitude test canned for overzealous conclusion.* Retrieved november 29th, 2016, from `http://retractionwatch.com/2014/07/18/the-camel-doesnt-have-two-humps-programming-aptitude-test-canned-for-overzealous-conclusion/`

Heijmeriks, G. (2007). *Opbouw lessenreeks.* Candea College.

Herman, G. L., Kaczmarczyk, L., Loui, M. C., & Zilles, C. (2008). Proof by incomplete enumeration and other logical misconceptions. In *Proceedings of the fourth international workshop on computing education research* (pp. 59–70).

Hickman, M., & Monaghan, J. (1993). *NETWORKING METHODOLOGIES: ISSUES ARISING FROM A RESEARCH STUDY EMPLOYING A MULTI-MEDIA ARTEFACT.*

Höffler, T. N., & Leutner, D. (2007). Instructional animation versus static pictures: A meta-analysis. , *17*, 722–738.

INFORMATICA-Actief, S. (2015). *Overzicht lesmateriaal.* Retrieved 6/3/2018, from `http://www.informatica-actief.nl/lesmateriaal.html`

Instruct. (2018). *Methode Informatica VO.* Retrieved 2018-03-06, from `http://instruct.nl/product/105/vo/fundament-informatica`

Kaczmarczyk, L. C., Petrick, E. R., East, J. P., & Herman, G. L. (2010). Identifying student misconceptions of programming. In *Proceedings of the 41st ACM technical symposium on computer science education - sigcse'10.* ACM Press.

Kuittinen, M., & Sajaniemi, J. (2004, sep). Teaching roles of variables in elementary programming courses. *ACM SIGCSE Bulletin*, *36*(3), 57. Retrieved from `http://dx.doi.org/10.1145/1026487.1008014` doi: 10.1145/1026487.1008014

Kurland, D. M., & Pea, R. D. (1985). Children's mental models of recursive LOGO programs. *Journal of Educational Computing Research*, *1*(2), 235–243.

Lahtinen, E., Ala-Mutka, K., & Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. In *Iticse* (Vol. 37, pp. 14–18). Monte de Caparica.

Ma, L., Ferguson, J., Roper, M., & Wood, M. (2007, mar). Investigating the viability of mental models held by novice programmers. *ACM SIGCSE Bulletin*, *39*(1), 499.

Ma, L., Ferguson, J., Roper, M., & Wood, M. (2011, mar). Investigating and improving the models of programming concepts held by novice programmers. *Computer Science Education*, *21*(1), 57–80.

Mayer, R. E. (1988). Introduction to research on teaching and learning computer programming. *Teaching and learning computer programming*, 1–12.

Mayer, R. E. (2002). Multimedia learning. In *Psychology of learning and motivation* (Vol. 41, pp. 85–139). Elsevier.

McComas, W. F. (2014). 21st-century skills. *The Language of Science Education*, *94*, 1. doi: 10.1007/978-94-6209-497-0

Miller, M. D., Linn, R. L., & Gronlund, N. E. (2009). *Measurement and assessment in teaching*. Pearson Higher Ed.

Pea, R. D. (1986). Language-independent conceptual "bugs" in novice programming. *Journal of Educational Computing Research*, *2*(1), 25–36.

Pea, R. D., Soloway, E., & Spohrer, J. C. (1987). The buggy path to the development of programming expertise. *Focus on Learning Problems in Mathematics*, *9*, 5–30.

Piaget, J., & Vygotskys, L. (1987). Theories on cognitive development. *Developmental psychology*.

Plass, D. (2015). *Identifying and Addressing Common Programming Misconceptions with Variables (Part 1)* . University of Twente.

Robins, A. (2010). Learning edge momentum: a new account of outcomes in CS1. *Computer Science Education*, *20*(1), 37–71.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, *13*(2), 137–172.

Schmidt, V. (2007). *Handreiking schoolexamen informatica havo/vwo.* Enschede: SLO.

Smith, J. P., DiSessa, A. A., & Roschelle, J. (1993). Misconceptions Reconceived: A Constructivistic Analysis of Knowledge in Transition. *Learning Sciences*, *3*(2), 115–163.

Soofos. (2015). *Hoe maak je de beste instructievideo's ?* Retrieved 2015-05-12, from `https://soofos.nl/hoe-maak-je-de-beste-instructievideos/`

Stichting Enigma Online. (2013). *De methode.* Retrieved 2018-03-06, from `https://www.enigma-online.nl/Page.aspx?pid=Methode`

Taylor, C., Zingaro, D., Porter, L., Webb, K., Lee, C., & Clancy, M. (2014). Computer science concept inventories: past and future. *Computer Science Education*, *24*(4), 253–276.

Tew, A. E. (2010). *Assessing fundamental introductory computing concept knowledge in a language independent manner* (Unpublished doctoral dissertation). Georgia Institute of Technology.

Tew, A. E., & Guzdial, M. (2011). The FCS1: a language independent assessment of CS1 knowledge. In *Proceedings of the 42nd acm technical symposium on computer science education* (pp. 111–116).

Thijs, A., Fisser, P., & van der Hoeven, M. (2014). *21e eeuwse vaardigheden in het curriculum van het funderend onderwijs.* Enschede: SLO.

Tolboom, J., Kruger, J., & Grgurina, N. (2014). *Informatica in de bovenbouw havo / vwo - Naar aantrekkelijk en actueel onderwijs in informatica* (Tech. Rep.). Enschede: SLO nationaal expertisecentrum leerplanontwikkeling.

Van der Meij, H., & Van der Meij, J. (2013). Eight Guidelines for the Design of Instructional Videos for Software Training. *Technical Communication*, *60*(3).

van der Meij, J., & van der Meij, H. (2015). A test of the design of a video tutorial for software training. *Journal of Computer Assisted Learning*, *31*(2), 116–132.

van Diepen, N. (2014). *11 redenen waarom programmeren zo moeilijk is.* Enschede.

Vygotsky, L. S. (1980). *Mind in society: The development of higher psychological processes*. Harvard university press.

Wilson, B. G. (2012). Constructivism in Practical and Historical Context. In R. A. Reiser & J. V. Dempsey (Eds.), *Trends and issues in instructional design and technology* (3th ed., pp. 45–52). Boston: Pearson.

Wilson, B. G. (2013). A Practice-centered Approach to Instructional Design. In M. M. Spector, B. B. Lockee, S. E. Smaldino, & M. Herring (Eds.), *Learning, problem solving, and mind tools* (pp. 1–19). Routledge.

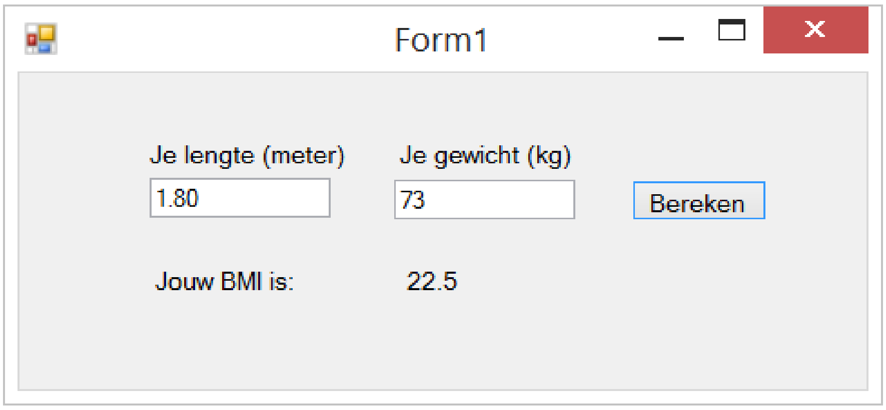
Winslett, G. (2014). What counts as educational video ?: Working toward best practice alignment between video production approaches and. *Australian Journal of Educational Technology*, *30*(5), 487–502.

Appendix A

BMI Assignment (Dutch)

A.1 Opdracht: BMI-Calculator

Maak een programma waarmee je de bodymass-index kan berekenen. De gebruiker geeft zijn gewicht in kilo en lengte in meter op. Vervolgens wordt met de bereken-knop de BMI uitgerekend en op het scherm getoond.



Tips:

- Maak eerst het formulier en voorzie de elementen van betekenisvolle namen.
- Voeg dan stap voor stap de code toe.
  1. Maak de variabelen aan voor de invoer.
  2. Haal de waardes op uit de tekstvelden en sla ze op in de variabelen.
  3. Maak nog een variabele aan en sla daarin het resultaat op van de BMI-berekening. (Zie `http://nl.wikipedia.org/wiki/Queteletindex` voor meer over BMI.)
  4. Laat het resultaat van de BMI-berekening op het scherm zien in een label.
  5. Controleer of je berekening klopt.

Wanneer je klaar bent, mag je het programma dummyproof maken:
Zorg ervoor dat de gebruiker geen negatieve waarde kan invoeren. Maak gebruik van keuzeopdrachten in VB met `If Then Else Endif`.
Bij een geldige invoer worden de berekeningen uitgevoerd. Bij een ongeldige invoer geef je een foutmelding met de functie `MsgBox()`.

Zorg ervoor dat gebruikers alleen getallen kunnen invoeren (dus geen letters of andere tekens). Maak hierbij in een keuzeopdracht gebruik van de functie `IsNumeric()`. Bij een geldige invoer worden de waardes uit de tekstvelden opgeslagen in de variabelen en wordt de rest van het programma uitgevoerd. Bij een ongeldige invoer geef je een foutmelding met de functie `MsgBox()`.

A.2 BMI Correction Model (Visual Basic)

The correct code for the BMI-assignment, following all guidelines and methods instructed in the classroom.

```
Public Class Form1
    Private Sub btnBereken_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles btnBereken.Click
        'declareren van variabelen
        Dim gewicht As Integer
        Dim lengte, bmi As Double

        'controleer of invoer numeriek is
        If IsNumeric(tbLengte.Text) And IsNumeric(tbGewicht.Text) Then
            'ophalen van waardes uit tekstvelden en opslaan in variabelen
            lengte = Val(tbLengte.Text)
            gewicht = Val(tbGewicht.Text)

            'controleer of invoer niet negatief is
            If lengte > 0 And gewicht > 0 Then

                'berekenen van BMI en resultaat opslaan in variabele
                bmi = gewicht / (lengte ^ 2)

                'extra: afronden van waarde naar 1 getal achter de komma
                bmi = Math.Round(bmi, digits:=1)

                'het resultaat van de berekening op het scherm laten zien
                lblResultaat.Text = Str(bmi)

            Else
                MsgBox("Lengte en gewicht mogen niet negatief zijn")
            End If

        Else
            MsgBox("Gewicht en lengte moeten een getal zijn")
        End If
    End Sub
End Class
```

A.3 Student's Code

Assignment as handed in by one of the students showing, amongst other mistakes, poor understanding and usage of variables by type conversion errors and non-usage of variables.

```
1  Public Class Form1
2      Private Sub btnberekenen_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnberekenen.Click
3          Dim lengte As Double
4          Dim gewicht As Double
5          Dim BMI As Double
6
7          lengte = Val(tblengte.Text)
8          gewicht = Val(tbgewicht.Text)
9
10         If tbLengte.Text > 0 Then
11
12             tbantwoord.Text = Val(gewicht)/(Val(lengte)*Val(lengte))
13
14         Else : MsgBox("U kunt geen negatieve getallen of letters gebruiken")
15         End If
16     End Sub
17 End Class
```

1. Line 10 - Improper (re)use of input elements and type mismatching; the If-operator requires first an argument evaluation to a Boolean value, the greater than operator (>) takes two numeric values to compare. The interface element `tbLengte.Text` contains a string, even though, the desired value has already been cast to a number and saved into the variable `lengte`.

2. Line 12 - Unfeasible direct output of computation results; common practise guidelines order to store results of data transformations in variables. A variable `BMI` has been declared but was not used to store the result of the computation and subsequently assign the value of `BMI` to the interface element `tbantwoord.Text`.

3. Line 12 - Incorrect type conversion; the interface element `tbantwoord.Text` requires a sting value, the function `Val()` requires a string argument. The variables `gewicht` and `length` already have numeric values. The function `Str()` was not used to convert the (computed) value before assigning it to `tbantwoord.Text`.

Appendix B

Visual Basic Tests (Dutch)

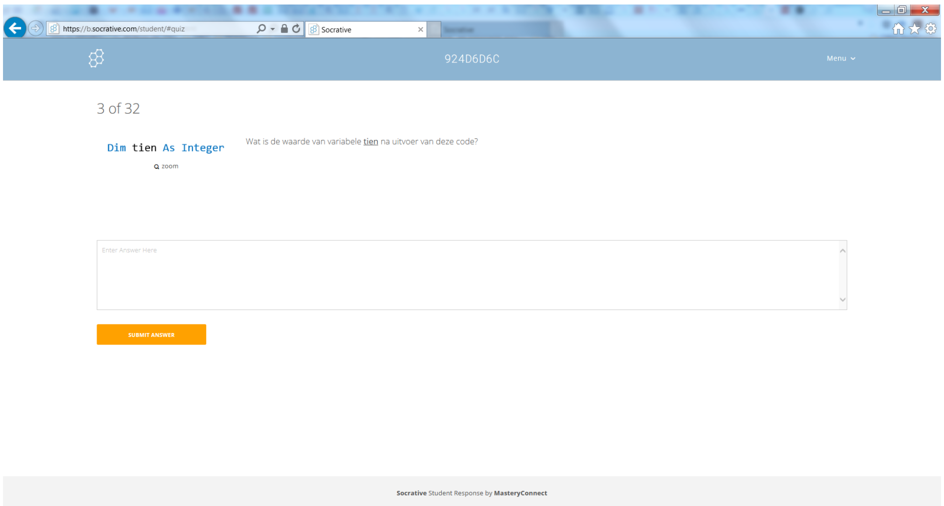


Figure B.1: screen of the pre-test in progress as displayed to students.

B.1 Pre-test

1. In welke klas zit je?

2. Heb je al eerder andere programmeertalen geleerd? Als dat zo is, welke talen ken je dan al?

3. Wat is de waarde van variabele `tien` na uitvoer van deze code?

```
Dim tien As Integer
```

4. Wat is de waarde van variabele `a` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
a = 7
b = a
```

5. Wat is de waarde van variabele `b` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
a = 7
b = a
```

6. Wat is de waarde van variabele `Hugo` na uitvoer van deze code?

```
Dim Hugo As Integer
Dim Tim As Integer
Hugo = 12
Tim = Hugo + 3
```

7. Wat is de waarde van variabele `Tim` na uitvoer van deze code?

```
Dim Hugo As Integer
Dim Tim As Integer
Hugo = 12
Tim = Hugo + 3
```

8. Wat is de waarde van variabele `x` na uitvoer van deze code?

```
Dim x As Integer
Dim y As Integer
x = 10
y = 20
x = y
```

9. Wat is de waarde van variabele `y` na uitvoer van deze code?

```
Dim x As Integer
Dim y As Integer
x = 10
y = 20
x = y
```

10. Wat is de waarde van variabele `groot` na uitvoer van deze code?

```
Dim groot As Integer
Dim klein As Integer
groot = 10
klein = 20
groot = klein
```

11. Wat is de waarde van variabele `klein` na uitvoer van deze code?

```
Dim groot As Integer
Dim klein As Integer
groot = 10
klein = 20
groot = klein
```

12. Wat is de waarde van variabele `a` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
a = 4
b = 3
b = a
```

13. Wat is de waarde van variabele `b` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
a = 4
b = 3
b = a
```

14. Wat is de waarde van variabele `i` na uitvoer van deze code?

```
Dim i As Integer
i = 1
i = i + 1
```

15. Wat is de waarde van variabele `x` na uitvoer van deze code?

```
Dim x As Integer
Dim y As Integer
y = 8
y = x + 10
```

16. Wat is de waarde van variabele `y` na uitvoer van deze code?

```
Dim x As Integer
Dim y As Integer
y = 8
y = x + 10
```

17. Van is de waarde van variabele `drie` na uitvoer van deze code?

```
Dim drie As Integer
drie = 5
```

18. Wat is de waarde van variabele `a` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
Dim c As Integer
a = 10
b = 20
c = 30
a = b
c = a
```

19. Wat is de waarde van variabele `b` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
Dim c As Integer
a = 10
b = 20
c = 30
a = b
c = a
```

20. Wat is de waarde van variabele `c` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
Dim c As Integer
a = 10
b = 20
c = 30
a = b
c = a
```

21. Wat is de waarde van variabele `a` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
a = 8
a = b * 4
```

22. Wat is de waarde van variabele `b` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
a = 8
a = b * 4
```

23. Wat is de waarde van variabele `a` na uitvoer van deze code?

```
Dim a As Integer
a = 10
a = a + 2
```

24. Wat is de waarde van variabele `a` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
a = 6
b = a + 1
```

25. Wat is de waarde van variabele `b` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
a = 6
b = a + 1
```

26. Wat is de waarde van variabele `x` na uitvoer van deze code?

```
Dim x As Integer
Dim y As Integer
x = 8
y = x
```

27. Wat is de waarde van variabele `y` na uitvoer van deze code?

```
Dim x As Integer
Dim y As Integer
x = 8
y = x
```

28. Wat is de waarde van variabele `straatnaam` na uitvoer van deze code?

```
Dim straatnaam As Integer
straatnaam = 101
```

29. Wat is de waarde van variabele `a` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
a = 10
b = 20
a = b
b = a
```

30. Wat is de waarde van variabele `b` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
a = 10
b = 20
a = b
b = a
```

31. Wat is de waarde van variabele `dozijn` na uitvoer van deze code?

```
Dim dozijn As Integer
```

32. Beschrijf wat er gebeurt op de laatste regel van deze code (`a = b`).

```
Dim a As Integer
Dim b As Integer
a = 9
b = 3
a = b
```

B.2 Post-test

1. Wat is de waarde van variabele `zes` na uitvoer van deze code?

```
Dim zes As Integer
zes = 15
```

2. Wat is de waarde van variabele `a` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
a = 100
b = 20
a = b
b = a
```

3. Wat is de waarde van variabele `b` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
a = 100
b = 20
a = b
b = a
```

4. Wat is de waarde van variabele `honderd` na uitvoer van deze code?

```
Dim honderd As Integer
```

5. Wat is de waarde van variabele `a` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
Dim c As Integer
a = 20
b = 40
c = 60
a = b
c = a
```

6. Wat is de waarde van variabele `b` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
Dim c As Integer
a = 20
b = 40
c = 60
a = b
c = a
```

7. Wat is de waarde van variabele `c` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
Dim c As Integer
a = 20
b = 40
c = 60
a = b
c = a
```

8. Wat is de waarde van variabele `x` na uitvoer van deze code?

```
Dim x As Integer
Dim y As Integer
x = 4
y = 10
x = y
```

9. Wat is de waarde van variabele `y` na uitvoer van deze code?

```
Dim x As Integer
Dim y As Integer
x = 4
y = 10
x = y
```

10. Wat is de waarde van variabele `a` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
a = 11
b = a
```

11. Wat is de waarde van variabele `b` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
a = 11
b = a
```

12. Wat is de waarde van variabele `paar` na uitvoer van deze code?

```
Dim paar As Integer
```

13. Wat is de waarde van variabele `i` na uitvoer van deze code?

```
Dim i As Integer
i = 4
i = i + 1
```

14. Wat is de waarde van variabele `a` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
a = 6
b = 8
b = a
```

15. Wat is de waarde van variabele `b` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
a = 6
b = 8
b = a
```

16. Wat is de waarde van variabele `achternaam` na uitvoer van deze code?

```
Dim achternaam As Integer
achternaam = 78
```

17. Wat is de waarde van variabele `x` na uitvoer van deze code?

```
Dim x As Integer
Dim y As Integer
y = 2
y = x + 20
```

18. Wat is de waarde van variabele `y` na uitvoer van deze code?

```
Dim x As Integer
Dim y As Integer
y = 2
y = x + 20
```

19. Wat is de waarde van variabele `maximum` na uitvoer van deze code?

```
Dim maximum As Integer
Dim minimum As Integer
maximum = 4
minimum = 100
maximum = minimum
```

20. Wat is de waarde van variabele `minimum` na uitvoer van deze code?

```
Dim maximum As Integer
Dim minimum As Integer
maximum = 4
minimum = 100
maximum = minimum
```

21. Wat is de waarde van variabele `Els` na uitvoer van deze code?

```
Dim Els As Integer
Dim Mirjam As Integer
Els = 2
Mirjam = Els + 23
```

22. Wat is de waarde van variabele `Mirjam` na uitvoer van deze code?

```
Dim Els As Integer
Dim Mirjam As Integer
Els = 2
Mirjam = Els + 23
```

23. Wat is de waarde van variabele `a` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
a = 10
a = b * 8
```

24. Wat is de waarde van variabele `b` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
a = 10
a = b * 8
```

25. Wat is de waarde van variabele `a` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
a = 10
b = a + 10
```

26. Wat is de waarde van variabele `b` na uitvoer van deze code?

```
Dim a As Integer
Dim b As Integer
a = 10
b = a + 10
```

27. Wat is de waarde van variabele `x` na uitvoer van deze code?

```
Dim x As Integer
Dim y As Integer
x = 30
y = x
```

28. Wat is de waarde van variabele `y` na uitvoer van deze code?

```
Dim x As Integer
Dim y As Integer
x = 30
y = x
```

29. Beschrijf wat er gebeurt op de laatste regel code.

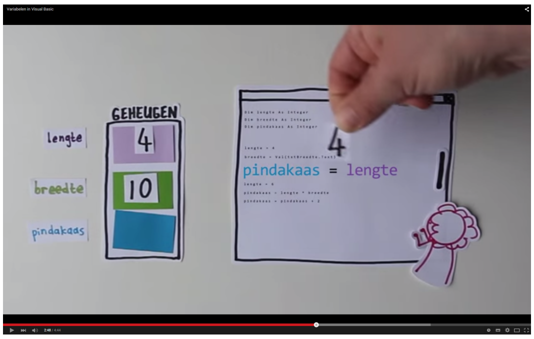
```
Dim a As Integer
Dim b As Integer
a = 16
b = 4
a = b
```

30. Nog vragen of opmerkingen?

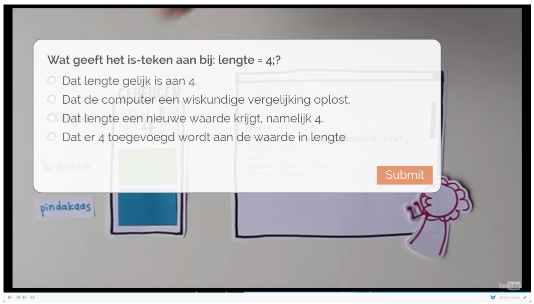
Appendix C

Interactive Video

The interactive video on Hapyak is no longer publicly available. The video—without questions—is available at `https://youtu.be/WARZCZ_D66Y`.



(a) memory-change animation



(b) multiple choice question

Figure C.1: stills of the interactive video.

C.1 Script (Dutch)

Opening Met variabelen kun je tijdelijk gegevens opslaan om later weer op te halen met de aangegeven variabelenaam. Hiermee kun je code leesbaarder en efficienter maken, en algemene oplossingen voor problemen programmeren waarvoor je de precieze gegevens nog niet weet — maar, *hoe werken variabelen*?

Introductie Nico Dit is architect Nico. Nico denkt slim te zijn en heeft zijn wachtwoord opgeslagen in de programmacode van zijn tekenprogramma.
Nico aan de telefoon: “Hai Anna! Ik heb m’n wachtwoord verstopt in een programma, in een variabele met de naam pindakaas. Slim he?”

Introductie Nellie En wie luistert daar het gesprek af? Dat is Nellie. Nellie is een hacker, en vindt het een interessante uitdaging om te zien of ze het wachtwoord kan vinden.

Van programma naar code Ze kijkt naar het scherm van het programma, maar daar ziet ze niet zo veel bijzonders aan. Dus ze gaat dieper en duikt in de code, op zoek naar *pindakaas*.

Declaratie Het programma begint met het aanmaken van een variabele. Om de gegevens in de variabele op te kunnen slaan reserveert de computer een stukje ruimte in het geheugen. Je geeft de variabele ook een naam. Die variabelenaam kun je daarna gebruiken om aan te geven dat je gegevens wilt opslaan op deze geheugenplek, of dat je de waarde van de gegevens op deze geheugenplek wilt bekijken. In dit programma wordt een variabele gemaakt met de naam *lengte*.

> **Quizvraag 1** Maakt het uit welke naam je kiest voor een variabele?
>
> a) Ja, want het bepaalt wat je erin kunt opslaan.
> b) Ja, de computer begrijpt waar de variabele voor is.
> c) Nee, maar het maakt de code wel leesbaarder.

Nu worden er nog wat variabelen aangemaakt. Hier een variabele met de naam *breedte* en daarna nog een met de naam *pindakaas*—hey pindakaas! Maar nu heeft pindakaas nog geen waarde gekregen.

> **Quizvraag 2** Wat is de waarde van `pindakaas` nu we het nog niet zelf een waarde hebben gegeven?
>
> a) De waarde van `pindakaas` is 0.
> b) De waarde van `pindakaas` is undefined (ongedefinieerd).
> c) De waarde van `pindakaas` kun je niet bekijken, je krijgt een Error.

Toekenning van een waarde Hier staat dat de variabele met naam *lengte* een nieuwe waarde krijgt, 4.

> **Quizvraag 3** Wat geeft het is-teken aan bij `lengte = 4`?
>
> a) Dat `lengte` gelijk is aan 4.
> b) Dat de computer een wiskundige vergelijking oplost.
> c) Dat `lengte` een nieuwe waarde krijgt, namelijk 4.
> d) Dat er 4 toegevoegd wordt aan de waarde in `lengte`.

Gebruikersinvoer De volgende regel ziet er wat ingewikkelder uit. De variabele *breedte* krijgt een nieuwe waarde, maar wat voor waarde? De waarde die staat in het tekstveld *txtBreedte*. Die waarde wordt gekopieerd in de geheugenplek van de variabele. Staat een variabelenaam voor het is-teken, dan betekent dat dus dat daar de nieuwe waarde naartoe moet. Staat een naam na het is-teken, dan wordt daar de waarde van opgevraagd.

Toekenning met andere variabele De waarde van variabele *lengte* wordt opgevraagd en een kopie wordt opgeslagen in de geheugenplek met de naam *pindakaas*. Nellie probeert het even uit, maar nee, dit is nog niet het juiste wachtwoord. Ze kijkt nog even verder.

> **Quizvraag 4** Wat gebeurt er als je de waarde van een variabele opslaat in een andere? (bijvoorbeeld `a = b`)
>
> a) De waarde van `b` wordt gekopieerd naar `a`.
> b) De waarde van `b` verplaatst naar `a`, `b` is nu leeg.
> c) De waardes van `a` en `b` worden omgewisseld.

Variabele *lengte* krijgt nu een nieuwe waarde. De oude waarde, 4, wordt overschreven met de nieuwe waarde 6.

> **Quizvraag 5** Wat gebeurt er als je een nieuwe waarde wilt opslaan in een variabele die al een waarde heeft? (bijvoorbeeld `a = 1; a = 2`)
>
> a) Dat kan niet. Een variabele kan maar 1 keer een waarde krijgen.
> b) De nieuwe waarde komt er bij, de variabele bevat de oude en de nieuwe waarde.
> c) De oude waarde verdwijnt, deze wordt vervangen door de nieuwe waarde.

Expressie Ah, hier gebeurt er weer iets met *pindakaas*. *Pindakaas* krijgt een nieuwe waarde. De waarde is een rekensom, lengte keer breedte. Eerst worden de waardes van de variabelen opgevraagd. *lengte* heeft waarde 6. *breedte* heeft waarde 10. De vermenigvuldiging wordt toegepast en het resultaat is 60. Dit wordt opgeslagen in *pindakaas*. Zou dit het wachtwoord zijn?—Nee, nog geen succes.

Toekenning met zichzelf in de expressie *Pindakaas* krijgt weer een nieuwe waarde—zou dit het zijn? *Pindakaas* krijgt de waarde van het resultaat van *pindakaas* plus 2. Eerst wordt weer de rekensom aan de rechterkant uitgerekend. De waarde van *pindakaas* is 60. Het resultaat van de som is dan 62. Dit wordt opgeslagen op de geheugenplek van *pindakaas*. *Pindakaas* is nu dus 62. Nellie probeert het uit, en ja, het werkt!

> **Quizvraag 6** Wat gebeurt er als je een berekening doet met een bepaalde variabele en je slaat het resultaat van die berekening op in dezelfde variabele? (bijvoorbeeld `a = a + b`)
>
> a) Dat geeft een foutmelding, de computer weet niet welke waarde hij moet gebruiken in de som.
> b) De berekening wordt uitgerekend met de huidige waarde van de variabele, het resultaat van de som wordt opgeslagen als de nieuwe waarde van de variabele.
> c) Dit wordt dan behandeld als een wiskundige vergelijking die de computer voor je oplost.

Slot Variabelen zijn dus namen voor geheugenplekken in de computer waar je tijdelijk gegevens in op kunt slaan om later weer op te vragen met de variabelenaam. Nu weet ook je hoe ze werken.

C.2 Program Code (Visual Basic)

```
Dim lengte As Integer
Dim breedte As Integer
Dim pindakaas As Integer

lengte = 4
breedte = Val(txtBreedte.Text)
pindakaas = lengte
lengte = 6
pindakaas = lengte * breedte
pindakaas = pindakaas + 2
```

C.3 Instruction (Dutch)

Participants received instructions about the study procedure and location of the assessment tests and video via ItsLearning, a common communication medium used by the school. For readability, the instruction text is presented below the picture.

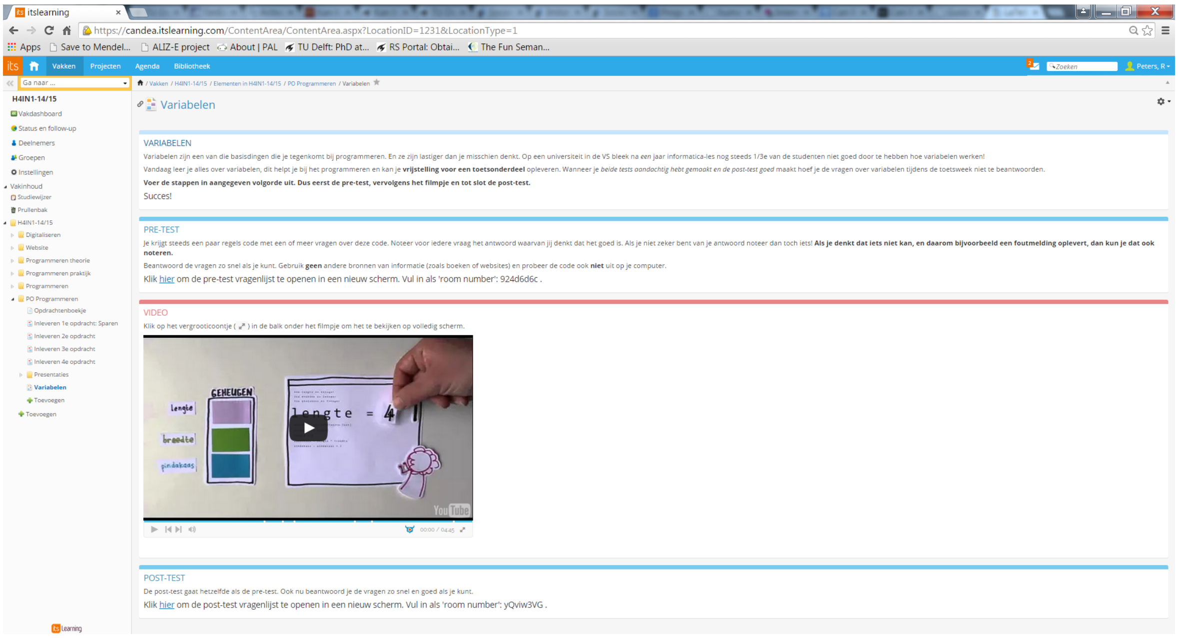


Figure C.2: instructions and hyperlinks as presented to the participants via ItsLearning.

Variabelen

Variabelen zijn een van die basisdingen die je tegenkomt bij programmeren. En ze zijn lastiger dan je misschien denkt. Op een universiteit in de VS bleek na een jaar informatica-les nog steeds 1/3e van de studenten niet goed door te hebben hoe variabelen werken!

Vandaag leer je alles over variabelen, dit helpt je bij het programmeren en kan je **vrijstelling voor een toetsonderdeel** opleveren. Wanneer je *beide tests aandachtig hebt gemaakt en de post-test goed* maakt hoef je de vragen over variabelen tijdens de toetsweek niet te beantwoorden.

Voer de stappen in de aangegeven volgorde uit. Dus eerst de pre-test, vervolgens het filmpje en tot slot de post-test.

Pre-test

Je krijgt steeds een paar regels code met een of meer vragen over deze code. Noteer voor iedere vraag het antwoord waarvan jij denkt dat het goed is. Als je niet zeker bent van je antwoord noteer dan toch iets! **Als je denkt dat iets niet kan, en daarom bijvoorbeeld een foutmelding oplevert, dan kun je dat ook noteren.**

Beantwoord de vragen zo snel als je kunt. Gebruik **geen** andere bronnen van informatie (zoals boeken of websites) en probeer de code ook **niet** uit op je computer.

Klik hier om de pre-test vragenlijst te openen in een nieuw scherm. Vul in als ‘room number’: 924d6d6c .

Video

Klik op het vergrooticoontje () in de balk onder het filmpje om het te bekijken op volledig scherm.

Post-test

De post-test gaat hetzelfde als de pre-test. Ook nu beantwoord je de vragen zo snel en goed als je kunt.

Klik hier om de post-test vragenlijst te openen in een nieuw scherm. Vul in als ’room number’: yQviw3VG .

Appendix D

Data

D.1 Raw Data

ppl#	Educational Level (Havo or VWO)	Programming Experience	3. tim (Dim tim As Integer)	4. a (Dim a As Integer = 7, Dim b As Integer = a)	5. b (Dim a As Integer = 7, Dim b As Integer = a)	6. Hugo (Dim Hugo As Integer = 12, Dim Tim As Integer = Hugo + 3)	7. Tim (Dim Hugo As Integer = 12, Dim Tim As Integer = Hugo + 3)	8. x (Dim x As Integer = 10, Dim y As Integer = 20, x = y)	9. y (Dim x As Integer = 10, Dim y As Integer = 20, x = y)	10. groot (Dim groot As Integer = 10, Dim klein As Integer = 20, groot = klein)	11. klein (Dim groot As Integer = 10, Dim klein As Integer = 20, groot = klein)	12. a (Dim a As Integer = 4, Dim b As Integer = 3, b = a)	13. b (Dim a As Integer = 4, Dim b As Integer = 3, b = a)	14. i (Dim i As Integer = 1, i = i + 1)	15. x (Dim x As Integer, Dim y As Integer = 8, y = x + 10)	16. y (Dim x As Integer, Dim y As Integer = 8, y = x + 10)	17. dde (Dim dde As Integer = 5)	18. a (Dim a As Integer = 10, Dim b As Integer = 20, Dim c As Integer = 30, a = b, c	19. b (Dim a As Integer = 10, Dim b As Integer = 20, Dim c As Integer = 30, a = b, c	20. c (Dim a As Integer = 10, Dim b As Integer = 20, Dim c As Integer = 30, a = b, c	21. a (Dim a As Integer = 8, Dim b As Integer, a = b * 4)	22. b (Dim a As Integer = 8, Dim b As Integer, a = b * 4)	23. a (Dim a As Integer = 10, a = a + 2)	24. a (Dim a As Integer = 6, Dim b As Integer = 7, a = b + 1)	25. b (Dim a As Integer = 6, Dim b As Integer = 7, a = b + 1)	26. x (Dim x As Integer = 8, Dim y As Integer = x)	27. y (Dim x As Integer = 8, Dim y As Integer = x)	28. straatnaam (Dim straatnaam As Integer = 101)	29. a (Dim a As Integer = 10, Dim b As Integer = 20, a = b, b = a)	30. b (Dim a As Integer = 10, Dim b As Integer = 20, a = b, b = a)	31. dodijn (Dim dodijn As Integer)
H00025B	Havo 4	None	10	7	7	12	15	Error	20	20	10	-1	1	2	-2	8	5	20	20	10	8	2	12	6	7	8	8	101	10	20	User input
H0003SZ	Havo 4	HTML	value assign	7	7	15	15	10	10	small	20	4	3	2	-2	8	5	10	20	30	8	2	12	6	7	8	8	101	10	20	no value
H0004BB	Havo 4	None	number	7	number	3	15	2	0,5	0,5	2	3/4	4/3	2	-2	0	5	1,5	0,5	2	8	2	8	8	7	8	8	0	4	2	number
H0005PK	Havo 4	None	0	7	7	12	15	20	20	20	20	4	3	2	0	X+10	5	20	20	20	can't multiply	0	12	6	7	8	8	101	20	20	0
H0006RB	Havo 4	Javascript, LUA, C#	value assign	7	7	12	15	20	20	20	20	4	4	2	No value	Error	5	20	20	20	b no value	no value	12	6	7	8	8	101	20	20	no value
H0007LK	Havo 4	Html, CSS, VB en	10	7	7	12	15	10	20	20	20	4	4	2	-2	8	5	20	20	20	2	2	10	6	7	8	8	101	20	20	no value
V0008CB	VWO 4	HTML, Visual Basics	10	49	7	12	15	2	0.5	2	0.5	-1	1	0	-2	8	5	2	0.5	1/3	2	2	8	6	7	8	8	101	Error	Error	12
V0009BV	VWO 4	None	number	7	a	hugo = tim - 3	15	10	20	2	0.5	-1	1	1+1	-2	x = 10	5	2	0.5	3	4b	2	8	6	7	8	8	101	20	10	Error
V0010SW	VWO 4	html	integer	a	a	12	15	10	20	10	Error	Error	3	2	-2	Error	5	20	20	10	b * 4	a/8*4	12	6	7	8	8	101	20	10	12
V0011MH	VWO 4	HTML	number	b	7	12	15	2	1	2	1	4	3	2	-2	10+x	5	2	20	30	4b	2	12	7	7	8-Y	8	101	2	2	no value
H0012JR	Havo 4	visual basic	number	7	a	12	15	10	20	10	20	4	3	1	2	8	5	10	20	30	8	2	8	6	7	8	8	101	10	20	12
H0013PV	Havo 4	None	10	7	7	12	15	Error	not equal	10	20	-1	-1	1	-2	8	5	10	20	30	64	2	10	7	7	8	8	101	20	20	12
H0014RH	Havo 4	None	10	7	7	12	15	10	20	20	20	4	4	2	-2	8	5	20	20	20	8	2	10	6	7	8	8	101	20	10	12
V0015JA	VWO 4	None	number	7	7	12	15	20	20	20	20	4	4	2	-2	8	5	20	20	20	8	2	12	6	7	8	8	101	20	20	unknown
H0016AZ	Havo 4	HTML, CSS en VB	number	7	7	12	15	10	20	Error	Error	Error	Error	Error	-2	8	5	20	10	Error	8	2	Error	7	7	8	8	101	10	10	no value
H0017KB	Havo 4	None	(heel getal)	7	7	9	15	2	30	200	20	12	12	2	-2	8	5	10	20	30	0	2	10	6	7	8	8	101	10	20	12
H0018JZ	Havo 4	None	number	7	7	12	15	10	20	20	20	4	4	2	-2	8	5	20	20	20	32	2	12	6	7	8	8	101	20	20	12
V0019MK	VWO 5	visual basic	value assign	7	7	12	15	20	20	20	20	3	3	2	-2	8	5	30	20	30	2	2	12	6	7	y	8	101	10	20	no value
V0020RH	VWO 4	None	value assign	7	7	12	15	20	20	20	20	4	4	2	No value	x = 10	5	20	20	20	b no value	no value	12	6	7	8	8	101	20	20	no value
H0021ND	Havo 4	None	number	7	7	15	15	20	20	20	20	4	a	2	x	10x	5	20	20	10	b*4	b	12	6	7	8	8	101	20	10	12
H0022EB	Havo 4	None	variable	7	7	12	15	equal thus	equal Error	error	Error	Error	Error	2	-2	8	5	Error	20	Error	32	variable	12	6	7	8	8	101	Error	Error	Variable
H0023UD	Havo 4	None	10	7	7	15	15	equal thus	not equal	not equal	not equal	Error	Error	2	-2	8	5	Error	Error	Error	8	2	10	6	7	8	8	101	Error	Error	12
H0024WW	Havo 4	None	Integer	7	7	12	15	10 and 20	10 and 20	10 or 20	10 or 20	3 or 4	3 or 4	2	-2	-2	5	10, 20 or 30	10 or 20	30 or 10	8	2	12	6	7	8	8	101	20 or 10	20 or 10	0
H0025BL	Havo 4	None	number	7	7	12	15	20	20	20	20	4	4	2	-2	10	5	20	20	20	8	0	12	6	7	8	8	101	20	10	no value
V0026DB	VWO 4	None	error	Error, a =7	7	12	15	10 and 20	30	20	30	7	7	1	-2	8	5	20	20	20	8	2	10	6	7	8	8	101	20	10	0
H0027LB	Havo 4	None	number	7	7	12	15	20	20	10	20	4	3	1	-2	8	5	10	20	30	8	2	10	6	5	8	8	101	10	20	12
V0029LL	VWO 4	html	error	7	7	12	15	10	20	20	20	4	4	V	-2	8	5	20	20	20	8	2	12	6	7	8	8	101	20	20	Error
H0030WH	Havo 4	None	number	7	7	12	15	20	20	20	20	4	4	2	-2	8	5	20	20	30	0	2	10	6	7	8	8	101	10	10	no value
V0031GP	VWO 4	None	no value	7	7	12	15	30	20	30	20	4	7	3	2	18+x	5	60	10	40	2	2	8	6	7	8	8	101	0	0	0
H0032JH	Havo 4	None	10	7	a	12	15	10	20	10	20	4	3	1	8	18	5	10	20	30	8	24	12	6	7	8	8	101	10	20	12
H0033SG	Havo 4	None	number	7	7	12	15	20	20	20	20	4	4	1	-2	18	5	20	20	10	8	2	12	6	7	8	8	101	20	10	number
H0034JN	Havo 4	None	number	7	7	12	15	20	20	20	20	4	4	2	-2	10	5	20	20	20	0	2	12	6	7	8	8	101	20	20	no value
V0035TB	VWO 4	html, visual basic	no value	7	7	12	15	30	30	30	20	4	4	2	-2	-2	5	20	10	30	8	2	10	6	7	8	8	101	20	10	number

Figure D.1: pre-test measures.

ppn	4. honderd (Dim honderd As Integer)	10. a (Dim a as Integer = 11, Dim b As Integer = a)	11. b (Dim a as Integer = 11, Dim b As Integer = a)	21. Eis (Dim Eis As Integer = 2, Dim Mijlam As Integer = Eis + 23)	22. Mijlam (Dim Eis As Integer = 2, Dim Mijlam As Integer = Eis + 23)	8. x (Dim x As Integer = 4, Dim y As Integer = 10, x = y)	9. y (Dim x As Integer = 4, Dim y As Integer = 10, x = y)	19. max (Dim max As Integer = 4, Dim min As Integer = 100, max = min)	20. min (Dim max As Integer = 4, Dim min As Integer = 100, max = min)	14. a (Dim a As Integer = 6, Dim b As Integer = 8, b = a)	15. b (Dim a As Integer = 6, Dim b As Integer = 8, b = a)	13. i (Dim i As Integer = 4, i = i + 1)	17. x (Dim x As Integer, Dim y As Integer = 2, y = x + 20)	18. y (Dim x As Integer, Dim y As Integer = 2, y = x + 20)	1. x (Dim six As Integer = 15)	5. a (Dim a As Integer = 20, Dim b As Integer = 40, Dim c As Integer = 60, …	6. b (Dim a As Integer = 20, Dim b As Integer = 40, Dim c As Integer = 60, …	7. c (Dim a As Integer = 20, Dim b As Integer = 40, Dim c As Integer = 60, …	23. a (Dim a As Integer = 10, Dim b As Integer, a = b * 8)	24. b (Dim a As Integer = 10, Dim b As Integer, a = b * 8)	25. a (Dim a As Integer = 10, Dim b As Integer = a + 10)	26. b (Dim a As Integer = 10, Dim b As Integer = a + 10)	27. x (Dim x As Integer = 30, Dim y As Integer = x)	28. y (Dim x As Integer = 30, Dim y As Integer = x)	16. voornaam (Dim voornaam As Integer = 78)	2. a (Dim a As Integer = 100, Dim b As Integer = 20, a = b, b = a)	3. b (Dim a As Integer = 100, Dim b As Integer = 20, a = b, b = a)	12. paar ((Dim paar As Integer)
H0002SB	Undefined	11	11	2	25	10	10	100	100	6	6	8	-22	2	15	40	40	40	Error	1.25	10	20	30	30	78	20	20	undefined
H0003SZ	no value	11	11	2	25	10	10	100	100	6	6	5	-18	2	15	40	40	20	10	1,25	10	20	30	30	78	20	100	no value assign
H0004BB	0	11	11	0	0	10	10	100	100	6	6	5	0	2	0	40	40	20	10	0	10	0	30	30	78	100	100	0
H0005PK	0	11	11	2	25	10	10	100	100	6	6	5	0	20	15	40	40	40	0	0	10	20	30	30	78	20	20	0
H0006RB	0	11	11	2	25	10	10	100	100	6	6	5	20	20	15	40	40	40	0	0	10	20	30	30	78	20	20	0
H0007LK	0	11	11	2	25	10	10	100	100	6	6	5	20	20	15	40	40	40	0	0	10	20	30	30	78	20	20	0
V0008CB	0	11	11	2	25	10	10	100	100	6	6	5	0	20	15	40	40	40	0	0	10	20	30	30	78	20	20	0
V0009BV	Error	11	11	2	25	10	10	100	100	6	6	5	x + 20	20	15	40	40	40	0	0	10	20	30	30	78	20	100	0
V0010SW	No value	11	11	2	25	10	10	100	100	6	6	5	Error	x + 20	15	40	40	40	b * 10	Error	10	20	30	30	78	20	100	no value assign
V0011MH	0	11	11	Error	25	10	10	100	100	6	6	5	-18	X+20	15	40	40	40	88	Error	10	20	30	30	78	20	20	0
H0012JR	0	10	11	2	21	4	4	4	100	6	8	4	-18	2	15	20	40	60	0	0	10	0	30	30	78	100	20	0
H0013PV	no value	11	11	2	25	10	10	100	100	8	8	5	22	22	15	40	20	60	0	0	10	20	30	30	78	20	20	0
H0014RH	0	11	11	2	25	10	10	100	100	6	6	5	-18	2	15	40	40	20	10	1.2	10	20	30	30	78	20	100	0
V0015JA	unknown	11	11	2	25	10	10	100	100	6	6	8	-18	2	15	40	40	40	10	Error	10	20	30	30	78	20	20	unknown
H0016AZ	0	11	11	2	25	10	10	100	100	6	6	5	0	20	15	40	40	20	0	0	10	20	30	30	78	20	100	0
H0017KB	0	11	11	2	25	10	10	100	100	6	6	5	0	20	15	40	40	40	0	0	10	20	30	30	78	20	20	0
H0018JZ	0	11	11	2	25	10	10	100	100	6	6	5	0	20	15	40	40	40	0	0	10	20	30	30	78	20	20	0
V0019MK	no value	11	11	2	25	10	10	100	100	6	6	5	unknown	x + 20	15	40	40	40	1,125	1,125	10	20	30	30	78	20	20	no value assign
V0020RH	0	11	11	2	25	10	10	100	100	6	6	5	0	20	15	40	40	40	0	0	10	20	30	30	78	20	20	0
H0021ND	0	11	11	2	25	10	10	100	100	6	6	5	0	20	15	40	40	40	b*8	0	10	20	30	30	78	20	20	0
H0022EB	0	11	11	2	25	10	10	100	100	6	6	5	0	20	15	40	40	40	0	0	10	20	30	30	78	20	20	0
H0023JD	0	11	11	2	25	10	10	100	100	6	6	5	-18	2	15	40	40	40	10	8	10	20	30	30	78	20	20	0
H0024WW	0	11	11	2	25	10	10	100	100	6	6	5	-18	2	15	40	40	20	10	0	10	20	30	30	78	100	20	0
H0025BL	0	11	11	2	25	10	10	100	100	6	6	5	0	20	15	40	40	20	0	0	10	20	30	30	78	20	100	0
V0026DB	undefined	11	11	2	25	10	10	100	100	6	6	5	0	20	15	40	40	20	80	0	10	20	30	30	78	20	100	undefined
H0027LB	no value	11	11	2	25	10	10	100	100	6	6	5	-18	2	15	40	40	40	10	1	10	20	30	30	78	20	20	no value
V0029LL	0	11	11	2	25	10	10	100	100	6	6	5	-18	2	15	40	40	20	10	1,25	10	20	30	30	78	20	100	0
H0030WH	0	11	11	2	25	10	10	100	100	6	6	5	20	20	15	40	40	40	0	0	10	20	30	30	78	20	20	0
V0031GP	0	11	11	2	25	10	10	4	4	6	6	5	y - 20	x + 20	15	60	20	60	b * 8	10/8	10	20	30	30	78	100	100	0
H0032JH	0	11	11	2	25	4	10	100	100	6	6	5	0	2	15	20	40	60	10	0.2	10	20	30	30	78	100	20	0
H0033SG	0	11	11	2	25	10	10	100	100	6	6	5	-18	-18	15	40	40	40	b * 8	1,25	10	20	30	30	78	20	20	0
H0034JN	no value	11	11	2	25	10	10	100	100	6	6	5	20	20	15	40	40	40	0	0	10	20	30	30	78	20	20	no value
V0035TB	no value	11	11	2	25	10	10	100	100	6	6	5	no value	22	15	40	40	20	80	563556	10	20	30	30	78	20	20	no value

Figure D.2: post-test measures.

D.2 Recoded and Transformed Data

candea_Catagorized_FINAL_DEF_AlphanumericCatagories(FINAL).sav

	pp#	Class	Experience	pre_a	pre_b	pre_c	pre_d	pre_e1	pre_e2	pre_f1	pre_f2	pre_g1	pre_g2	pre_h1
1	H0002SB	0	0	H2	misc	0	0	H2S3	0	0	0	0	0	M2
2	H0003SZ	0	0	E	E	0	0	0	C	misc	0	0	0	0
3	H0004BB	0	0	E	E	0	H1	misc	misc	misc	0	0	E	M1
4	H0005PK	0	0	0	0	0	0	0	0	0	0	0	0	0
5	H0006RB	0	1	E	E	0	0	0	0	0	0	0	0	0
6	H0007LK	0	0	H2	E	0	0	0	0	0	0	0	0	0
7	V0008CB	1	0	H2	H2	0	0	M2D	M2	0	0	misc	0	M2
8	V0009BV	1	0	E	D	0	0	M2D	M2	C	0	0	C	M2
9	V0010SW	1	0	E	H2	0	0	H1S1	H1S1	0	0	C	C	S1
10	V0011MH	1	0	E	E	0	0	misc	M2	0	0	C	0	0
11	H0012JR	0	0	E	H2	0	0	0	M3S1	0	0	0	C	0
12	H0013PV	0	0	H2	H2	0	0	0	M3S1	0	0	0	0	M2
13	H0014RH	0	0	H2	H2	0	0	0	0	0	0	0	0	0
14	V0015JA	1	0	E	E	0	0	0	0	0	0	0	0	0
15	H0016AZ	0	0	E	E	0	0	S1	S1	0	0	0	0	S1
16	H0017KB	0	0	E	H2	0	0	0	misc	misc	0	0	0	misc
17	H0018JZ	0	0	E	H2	0	0	0	0	0	0	0	0	0
18	V0019MK	1	0	E	E	0	0	0	0	0	0	0	0	S2
19	V0020RH	1	0	E	E	0	0	0	0	0	0	0	0	0
20	H0021ND	0	0	E	H2	0	0	0	0	misc	0	0	0	0
21	H0022EB	0	0	misc	misc	0	0	S1	S1	0	0	0	0	S1
22	H0023JD	0	0	H2	H2	0	0	S1	S1	misc	0	0	0	S1
23	H0024WW	0	0	E	0	0	0	M1	M1	0	0	0	0	M1
24	H0025BL	0	0	E	E	0	0	0	0	0	0	0	0	0
25	V0026DB	1	0	D	0	0	0	misc	0	0	0	S1	0	misc
26	H0027LB	0	0	E	H2	0	0	0	M3S1	0	0	0	0	0
27	V0029LL	1	0	D	D	0	0	0	0	0	0	0	0	0
28	H0030WH	0	0	E	E	0	0	0	0	0	0	0	0	0
29	V0031GP	1	0	E	0	0	0	0	S4	0	0	0	0	0
30	H0032JH	0	0	H2	H2	0	0	0	M3S1	0	0	0	C	0
31	H0033SG	0	0	E	E	0	0	0	0	0	0	0	0	0
32	H0034JN	0	0	E	E	0	0	0	0	0	0	0	0	0
33	V0035TB	1	0	E	E	0	0	0	S4	0	0	0	0	0

candea_Catagorized_FINAL_DEF_AlphanumericCatagories(FINAL).sav

	pre_h2	pre_i1	pre_i2	pre_j1	pre_j2	pre_k1	pre_k2	pre_l	pre_m1	pre_m2	pre_n1	pre_n2	pre_o1	pre_o2
1	M2	S1	0	M2	M2	M2	M2	0	0	0	0	0	0	0
2	M3S1	S2	S2	M2	M2	M2	M2	0	0	0	0	0	M3S1	0
3	M1	M2	M2D	M2	misc	M2	M2	0	misc	0	0	0	D	M2D
4	M3S1	0	0	0	C	misc	0	0	0	0	0	0	0	0
5	0	0	0	E	E	E	E	0	0	0	0	0	0	0
6	0	M3S1	0	M2	M2	misc	M2	0	0	0	0	0	0	0
7	M2	M2	M2D	M2	M2	misc	M2	M1	0	0	0	0	M2	M2D
8	M2	M3S1	0	M2	C	C	M2	C	0	0	0	0	M2	M2D
9	S1	M3S1	0	M2	E	C	C	0	0	0	0	0	0	0
10	M3S1	M2	M2	M2	C	C	M2	0	misc	0	C	0	M2	0
11	M3S1	M3S1	0	M2	M2	M2	M2	M3S1	0	0	0	0	M3S1	0
12	misc	S1	S1	M2	M2	misc	M2	M3S1	misc	0	0	0	M3S1	0
13	0	M3S1	0	M2	M2	M2	M2	0	0	0	0	0	0	0
14	0	0	0	M2	M2	M2	M2	0	0	0	0	0	0	0
15	S1	M3S1	0	M2	M2	M2	M2	M2S5	misc	0	0	0	0	S4
16	misc	M2	M1	M2	M2	0	M2	0	0	0	0	0	M3S1	0
17	0	M3S1	0	M2	M2	misc	M2	0	0	0	0	0	0	0
18	S2	0	0	M2	M2	misc	M2	0	0	0	C	0	S4	0
19	0	0	0	E	C	E	E	0	0	0	0	0	0	0
20	C	0	0	C	C	C	C	0	0	0	0	0	0	0
21	S1	S1	S1	M2	M2	misc	misc	0	0	0	0	0	S1	0
22	S1	S1	S1	M2	M2	M2	M2	0	0	0	0	0	S1	S1
23	M1	C2	M1	M2	misc	M2	M2	0	0	0	0	0	M1	M1
24	0	0	0	M2	0	M3S1	0	0	0	0	0	0	0	0
25	S4	C2	M1	M2	M2	M2	M2	M3S1	0	0	0	0	0	0
26	M3S1	0	0	M2	M2	M2	M2	M3S1	0	misc	0	0	M3S1	0
27	0	M3S1	0	M2	M2	M2	M2	0	0	0	0	0	0	0
28	0	0	0	M2	M2	0	M2	0	0	0	0	0	0	0
29	S4	S4	0	misc	C	misc	M2	S4	0	0	0	0	misc	S3
30	M3S1	M3S1	0	S1	S4	misc	misc	M3S1	0	0	0	0	M3S1	0
31	0	0	0	M2	S4	M2	M2	M3S1	0	0	0	0	0	0
32	0	0	0	M2	0	0	M2	0	0	0	0	0	0	0
33	0	S4	M1	M2	misc	M2	M2	0	0	0	0	0	0	S3

	pre_o3	pre_p1	pre_p2	post_a	post_b	post_c	post_d	post_e1	post_e2	post_f1	post_f2	post_g1	post_g2	post_h1
1	ID	M3S1	0	E	E	0	0	0	0	0	0	0	0	0
2	M3S1	M3S1	0	E	E	0	0	0	0	0	0	0	0	0
3	misc	misc	M2	0	0	H1	0	0	0	misc	misc	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	misc	S1	S1	0	0	0	0	0	0	0	0	0	0	0
8	M2	0	ID	D	0	0	0	0	0	0	0	0	0	0
9	ID	0	ID	E	E	0	0	0	0	0	0	0	0	0
10	misc	misc	M2	0	0	0	0	0	0	H1S1	0	0	0	0
11	M3S1	M3S1	0	0	0	0	0	0	M3S1	0	misc	misc	0	0
12	M3S1	0	0	E	0	0	0	0	0	0	0	0	0	S2
13	0	0	ID	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	E	E	0	0	0	0	0	0	0	0	0
15	S1	S2	S2	0	0	0	0	0	0	0	0	0	0	0
16	M3S1	M3S1	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	misc	M3S1	0	E	E	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	ID	0	ID	0	0	0	0	0	0	0	0	0	0	0
21	S1	S1	S1	0	0	0	0	0	0	0	0	0	0	0
22	S1	S1	S1	0	0	0	0	0	0	0	0	0	0	0
23	M1	M1	M1	0	0	0	0	0	0	0	0	0	0	0
24	0	0	ID	0	0	0	0	0	0	0	0	0	0	0
25	0	0	ID	E	E	0	0	0	0	0	0	0	0	0
26	M3S1	M3S1	0	E	E	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	misc	S2	S2	0	0	0	0	0	0	0	0	0	0	0
29	misc	misc	misc	0	0	0	0	S2	S2	0	0	0	0	0
30	M3S1	M3S1	0	0	0	0	0	0	0	0	0	0	0	0
31	ID	0	ID	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	E	E	0	0	0	0	0	0	0	0	0
33	misc	0	ID	E	E	0	0	0	0	0	0	0	0	0

candea_Catagorized_FINAL_DEF_AlphanumericCatagories(FINAL).sav

	post_h2	post_i1	post_i2	post_j1	post_j2	post_k1	post_k2	post_l	post_m1	post_m2	post_n1	post_n2	post_o1	post_o2
1	0	0	0	misc	M3S1	E	M2D	S4	0	0	0	0	0	0
2	0	0	0	M2	M2	M2	M2D	0	0	0	0	0	0	0
3	0	0	0	0	M3S1	M2	0	0	0	S1	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	misc	0	0	0	0	0	0	0	0	0	0
6	0	0	0	misc	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	C	0	0	0	0	0	0	0	0	0	0
9	0	0	0	E	C	C	E	0	0	0	0	0	0	0
10	0	0	0	M2	C	C	E	0	0	0	0	0	0	0
11	M3S1	S2	S2	M2	M2	0	0	M3S1	0	S1	0	0	M3S1	0
12	S2	0	0	misc	misc	0	0	0	0	0	0	0	0	S3
13	0	0	0	M2	M2	M2	M2D	0	0	0	0	0	0	0
14	0	0	0	M2	M2	M2	E	S4	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	E	C	misc	M2D	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	C	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	M2	M2	misc	misc	0	0	0	0	0	0	0
23	0	0	0	M2	M2	M3S1	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	misc	0	0	0	0	0	0	0	0
26	0	0	0	M2	M2	M2	M2	0	0	0	0	0	0	0
27	0	0	0	M2	M2	M2	M2D	0	0	0	0	0	0	0
28	0	0	0	misc	0	0	0	0	0	0	0	0	0	0
29	0	0	0	C	C	C	C	0	0	0	0	0	misc	S3
30	0	M3S1	0	0	M3S1	M2	M2D	0	0	0	0	0	M3S1	0
31	0	0	0	M2	misc	C	M2D	0	0	0	0	0	0	0
32	0	0	0	misc	0	0	0	0	0	0	0	0	0	0
33	0	0	0	E	misc	misc	misc	0	0	0	0	0	0	0

candea_Catagorized_FINAL_DEF_AlphanumericCatagories(FINAL).sav

	post_o3	post_p1	post_p2	PreSum ofErrors	PostSum ofErrors	PreSumofMis conceptions	PostSumofMi sconceptions	PreSumofPri orknowledge	PostSumofPri orknowledge	PreSum ofOther
1	0	0	0	12	7	10	2	0	4	2
2	ID	0	ID	14	8	10	3	3	3	1
3	ID	S2	S2	21	9	8	6	6	0	7
4	0	0	0	3	0	1	0	1	0	1
5	0	0	0	6	1	0	0	6	0	0
6	0	0	0	7	1	5	0	1	0	1
7	0	0	0	19	0	13	0	3	0	3
8	0	0	ID	18	3	8	0	9	2	1
9	0	0	ID	15	7	7	0	6	6	2
10	0	0	0	18	5	8	2	6	3	4
11	M3S1	M3S1	0	14	13	12	11	2	0	0
12	misc	0	0	15	7	12	3	0	1	3
13	ID	0	ID	8	6	7	3	0	1	1
14	0	0	0	6	7	4	4	2	3	0
15	ID	0	ID	17	2	14	0	2	0	1
16	0	0	0	14	0	9	0	1	0	4
17	0	0	0	7	0	5	0	1	0	1
18	0	0	0	12	6	7	0	3	5	2
19	0	0	0	6	0	0	0	6	0	0
20	0	0	0	10	1	1	0	6	1	3
21	0	0	0	16	0	12	0	0	0	4
22	0	0	0	18	4	17	2	0	0	1
23	ID	M3S1	0	16	5	14	4	1	0	1
24	ID	0	ID	5	2	2	0	2	0	1
25	ID	0	ID	13	5	9	0	1	2	3
26	0	0	0	13	6	11	4	1	2	1
27	ID	0	ID	7	6	5	3	2	1	0
28	0	0	0	8	1	5	0	2	0	1
29	misc	S2	S2	14	11	6	5	2	4	6
30	M3S1	M3S1	0	14	7	11	6	1	1	2
31	0	0	0	9	4	5	1	2	2	2
32	0	0	0	4	3	2	0	2	2	0
33	ID	0	0	12	7	7	0	2	3	3

candea_Catagorized_FINAL_DEF_AlphanumericCatagories(FINAL).sav

	PostSumofOther	PreScoreH2	PostScoreH2	PreScoreM1	PostScoreM1	PreScoreM2	PostScoreM2	PreScoreM3S1	PostScoreM3S1	PreScoreS1	PostScoreS1
1	1	2	0	0	0	6	0	1	1	1	0
2	2	0	0	0	0	4	3	4	0	0	0
3	3	0	0	2	0	5	1	0	1	0	1
4	0	0	0	0	0	0	0	1	0	0	0
5	1	0	0	0	0	0	0	0	0	0	0
6	1	1	0	0	0	3	0	1	0	0	0
7	0	2	0	1	0	8	0	0	0	2	0
8	1	0	0	0	0	7	0	1	0	0	0
9	1	1	0	0	0	1	0	1	0	4	0
10	0	0	0	0	0	7	1	1	0	0	1
11	2	1	0	0	0	4	2	7	6	0	1
12	3	2	0	0	0	4	0	4	0	2	0
13	2	2	0	0	0	4	3	1	0	0	0
14	0	0	0	0	0	4	3	0	0	0	0
15	2	0	0	0	0	5	0	1	0	5	0
16	0	1	0	1	0	4	0	3	0	0	0
17	0	1	0	0	0	3	0	1	0	0	0
18	1	0	0	0	0	3	0	1	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0
20	0	1	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	2	0	0	0	10	0
22	2	2	0	0	0	4	2	0	0	11	0
23	1	0	0	10	0	3	2	0	2	0	0
24	2	0	0	0	0	1	0	1	0	0	0
25	3	0	0	1	0	4	0	1	0	1	0
26	0	1	0	0	0	4	4	6	0	0	0
27	2	0	0	0	0	4	3	1	0	0	0
28	1	0	0	0	0	3	0	0	0	0	0
29	2	0	0	0	0	1	0	0	0	0	0
30	0	2	0	0	0	0	1	7	5	1	0
31	1	0	0	0	0	3	1	1	0	0	0
32	1	0	0	0	0	2	0	0	0	0	0
33	4	0	0	1	0	3	0	0	0	0	0

candea_Catagorized_FINAL_DEF_AlphanumericCatagories(FINAL).sav

	PreScore S2	PostScoreS2	PreScore S4	PostScore S4	PreScoreH2 Cluster	PostScoreH 2Cluster	PreScoreM 1Cluster	PostScoreM 1Cluster	PreScoreM 2Cluster
1	0	0	0	1	1	0	0	0	4
2	2	0	0	0	0	0	0	0	4
3	0	2	0	0	0	0	2	0	3
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	1	0	0	0	3
7	0	0	0	0	2	0	0	0	3
8	0	0	0	0	0	0	0	0	2
9	0	0	0	0	1	0	0	0	1
10	0	0	0	0	0	0	0	0	2
11	0	2	0	0	1	0	0	0	4
12	0	2	0	0	2	0	0	0	3
13	0	0	0	0	2	0	0	0	4
14	0	0	0	1	0	0	0	0	4
15	2	0	1	0	0	0	0	0	4
16	0	0	0	0	1	0	1	0	3
17	0	0	0	0	1	0	0	0	3
18	2	0	1	0	0	0	0	0	3
19	0	0	0	0	0	0	0	0	0
20	0	0	0	0	1	0	0	0	0
21	0	0	0	0	0	0	0	0	2
22	0	0	0	0	2	0	0	0	4
23	0	0	0	0	0	0	5	0	3
24	0	0	0	0	0	0	0	0	1
25	0	0	1	0	0	0	1	0	4
26	0	0	0	0	1	0	0	0	4
27	0	0	0	0	0	0	0	0	4
28	2	0	0	0	0	0	0	0	3
29	0	4	4	0	0	0	0	0	1
30	0	0	1	0	2	0	0	0	0
31	0	0	1	0	0	0	0	0	3
32	0	0	0	0	0	0	0	0	2
33	0	0	2	0	0	0	1	0	3

candea_Catagorized_FINAL_DEF_AlphanumericCatagories(FINAL).sav

	PostScoreM2Cluster	PreScoreM3Cluster	PostScoreM3Cluster	PreScoreS1Cluster_notused	PostScoreS1Cluster_notused	PreScoreS1eCluster	PostScoreS1eCluster	PreScoreS2Cluster	PostScoreS2Cluster
1	0	0	0	0	0	0	0	0	0
2	3	1	0	0	0	0	0	2	0
3	1	0	0	0	1	0	1	0	0
4	0	1	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	1	0	0	0	0	0	0	0
9	0	1	0	0	0	0	0	0	0
10	1	1	0	0	1	0	0	0	0
11	2	3	2	0	1	0	1	0	2
12	0	1	0	0	0	0	0	0	2
13	3	1	0	0	0	0	0	0	0
14	3	0	0	0	0	0	0	0	0
15	0	1	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0
17	0	1	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	2	0
19	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0
22	2	0	0	0	0	0	0	0	0
23	2	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0
26	4	2	0	0	0	0	0	0	0
27	3	1	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	2
30	1	3	1	0	0	0	0	0	0
31	1	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0

candea_Catagorized_FINAL_DEF_AlphanumericCatagories(FINAL).sav

	PreScoreS4Cluster	PostScoreS4Cluster	PreScore1	PostScore1	PreScore2	PostScore2	PreScore2e	PostScore2e	PreScore3	PostScore3
1	0	1	2	0	7	0	8	1	0	0
2	0	0	0	0	6	3	10	3	0	0
3	0	0	1	1	7	4	7	5	0	0
4	0	0	0	0	0	0	1	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	1	0	3	0	4	0	0	0
7	0	0	2	0	11	0	11	0	0	0
8	0	0	0	0	7	0	8	0	0	0
9	0	0	1	0	5	0	6	0	0	0
10	0	0	0	0	7	2	8	2	0	0
11	0	0	1	0	4	5	11	11	0	0
12	0	0	2	0	6	2	10	2	0	1
13	0	0	2	0	4	3	5	3	0	0
14	0	1	0	0	4	3	4	3	0	0
15	1	0	0	0	12	0	13	0	0	0
16	0	0	1	0	5	0	8	0	0	0
17	0	0	1	0	3	0	4	0	0	0
18	1	0	0	0	5	0	6	0	0	0
19	0	0	0	0	0	0	0	0	0	0
20	0	0	1	0	0	0	0	0	0	0
21	0	0	0	0	12	0	12	0	0	0
22	0	0	2	0	15	2	15	2	0	0
23	0	0	0	0	13	2	13	4	0	0
24	0	0	0	0	1	0	2	0	0	0
25	0	0	0	0	6	0	7	0	0	0
26	0	0	1	0	4	4	10	4	0	0
27	0	0	0	0	4	3	5	3	0	0
28	0	0	0	0	5	0	5	0	0	0
29	1	0	0	0	1	4	1	4	1	1
30	0	0	2	0	1	1	8	6	0	0
31	0	0	0	0	3	1	4	1	0	0
32	0	0	0	0	2	0	2	0	0	0
33	0	0	0	0	4	0	4	0	1	0

candea_Catagorized_FINAL_DEF_AlphanumericCatagories(FINAL).sav

	PreScore4	PostScore 4	PreScore5	PostScore 5	PreScore6	PostScore 6
1	1	1	0	1	0	0
2	4	0	0	0	0	0
3	0	1	0	0	0	0
4	1	0	0	0	0	0
5	0	0	0	0	0	0
6	1	0	0	0	0	0
7	0	0	0	0	0	0
8	1	0	0	0	0	0
9	1	0	0	0	0	0
10	1	0	0	0	0	0
11	7	6	0	0	0	0
12	4	0	0	0	0	0
13	1	0	0	0	0	0
14	0	0	0	1	0	0
15	1	0	1	0	1	0
16	3	0	0	0	0	0
17	1	0	0	0	0	0
18	1	0	1	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	2	1	0	0	0
24	1	0	0	0	0	0
25	1	0	2	0	0	0
26	6	0	0	0	0	0
27	1	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	4	0	0	0
30	7	5	1	0	0	0
31	1	0	1	0	0	0
32	0	0	0	0	0	0
33	0	0	2	0	0	0