

Improving the Flipped Classroom Perspective for Programming in Creative Technology

Jur van Geel
Creative Technology
Ansgar Fehnker
Angelika Mader
15th of February 2019

ABSTRACT

This thesis researches both the need for library education for the creative technology curriculum and its implementation. It tests the already existing Flipped classroom model and the newly designed Guided Read-in as methodologies for this implementation. It does so by firstly stating the need for library education within the Creative Technology curriculum. Afterwards the State-of-the-Art regarding introductory programming education and libraries in the context of the OpenFrameworks platform is reported. The second part evaluates both methodologies as they were implemented within the Programming for AI 2019 course.

Both the OpenCV library and OFxGui play a major part in this research. OpenCV was used as library for the Guided Read-in and OFxGui was introduced using the Flipped Classroom approach. Both methodologies are suited for completely different purposes and both exist within the context of introductory programming education.

Regarding the Flipped classroom principle, conclusions are drawn about its suitedness for academic purposes. When introducing students to the Flipped classroom approach, the teacher should keep in mind that deadlines help students to keep on track. As not all students react toward the Flipped classroom approach in the same manner, many require more regulation. When introducing Flipped classroom in the academic curricula teacher need to take a good look-out for which teaching method is the most suitable and how they want to implement this in their standing methodology.

ACKNOWLEDGEMENT

This research wouldn't have been possible without the amazing help received by the Creative Technology staff. Their interest and compassion have been absolutely essential, their willingness to assist is what brought this piece to life. Especially, I want to thank Karin Slotman, Robby van Delden and Jasper Goseling for their active role in the evaluation that has been reported in this thesis.

I would also like to thank Marcus Gerhold, Chris Zeinstra and the pool of teaching assistance for their participation in organizing the course Programming for AI 2018. Without the wonderful job of carrying the course another year forward I would have been able to conduct my thesis in any form. Especially Dennis Vinke has been there for me every step of the way as a sparring partner and oracle. But a course is not complete without the students taking the course. Which makes me endlessly grateful for the Creative Technology class of 2020 for being my Guinea pigs.

I'd also like to thank Lewis Lepton for his invaluable line of tutorial on YouTube. Not only have you allowed me to test the Flipped Classroom principle on OFxGui but also helped many a student pass the course Programming for AI 2019

Finally, and most importantly, thank you Ansgar and Angelika. Although we may never completely see eye to eye, without the both of you this thesis would have never seen the light of day. In providing feedback, guiding me through the woods and critically sending me back when work was not good enough you have made me rise above my station. It's all this that has brought me to here and I am very proud of the work that has become my thesis.

TABLE OF CONTENTS

Abstract	2
Acknowledgement.....	3
Table of Contents	4
Chapter 1 - Introduction	6
Chapter 2 – Method.....	8
Chapter 3 – State of Creative Technology	10
What is Creative Technology.....	10
State of Creative Technology - Staff interviews	11
State of Creative Technology - Programming	15
state of Creative Technology - Conclusion	16
Chapter 4 – State of the Art on Programming education.....	17
High failure ratings in introductory programming courses	17
Key factors for introductory programming courses	18
Teaching methods for introductory programming courses	20
<i>Examples of teaching methods</i>	21
Student Engagement	22
Students Ownership	22
The Flipped Classroom.....	23
Chapter 5 – State of the Art on Libraries (for OpenFrameworks and processing)	24
Graphics	25
OFxSVG	25
OFxAssimpModelLoader	25
OpenGL.....	25
OFxGui.....	26
Mobile devices	27
OFxAccelerometer	27
Android.....	27
iOS	28
Web / Network	28
OFxEmscripten	28
OFxXmlSettings	29
OFxNetwork	29
OFxOsc (Open Sound Control)	29

OFxPoco	30
Vision	30
OFxKinect	30
OFxOpenCV	30
Chapter 6 – Realization	32
Specification	32
Implementation of the tutorials into the 2018 Course	33
Prototyped Component	35
The added tutorials	35
Evaluation Rubric	37
Chapter 7 – Implementation	38
Exploration of the Libraries	38
Chapter 8 – Results	40
Students	40
Questionnaires	40
Evaluation session	41
Use of libraries in end-assignments	43
Staff	45
Chapter 9 Conclusions	46
Performance of the designed courses	46
Chapter 10 Future work	48
feedback On the course Programming for AI	48
References	50
Appendix A through N	52
A - Staff Interviews	52
B – Evaluation Rubric	59
c – OFxGui tutorial / Flipped Classroom	60
D – OpenCV Tutorial / Guided read-in	61
e – SIMS motivational Scale	66
F – Questionnaire 1	67
G – Questionnaire 2	71
H – Results Questionnaire 1	75
I – Results Questionnaire 2	86
J – Statistical analysis	97

CHAPTER 1 - INTRODUCTION

Creative Technology (also called CreaTe) is an English-taught programme which combines computer science and electrical engineering in order to create new solutions with real impact on people. It embodies the University of Twente's motto; "High tech, human touch". It is the study for the new engineer, a T-shaped professional, who maintained a broad view of the world during his bachelor for him to stay multidisciplinary. This multidisciplinary approach is necessary to keep up with the speed of technological development within our modern world. It is truly a digital engineering study for future professionals.

Within modern engineering studies, it is hardly unthinkable to not cover the basics of programming. Hence is programming also one of the pillars within Creative technology. More and more fields require students to be able to work with, understand and create algorithms out of thin air. This ever-growing demand asks of the educational system almost everyone possess the ability to write up new algorithms from thin air. Yet learning to basically speak computer as a second language has been known to be very difficult due to the lack of time to perfect the art (Cunningham, Sanjuan Espejo, Frederick, Sun, & Ding, 2016). In order to 'speak' algorithmic a lot of logic thinking, and problem-solving skills are required and creating these insides takes a tremendous amount of time and effort.

Creative Technology is still a young and diverse study with an open mindset and a continues strive for innovation. In this study, which is constantly trying to improve not only the world but also its own curriculum, it is essential that students learn the foundations of algorithmic thought. But with a large diversity in student backgrounds it is important to consider the scale in student differences. As mentioned before, also this engineering program cannot turn a blind eye to teaching programming within its curriculum. But how can this algorithmic thinking best be educated to the new Creative Technologist?

In order to continue improving the curriculum this research focusses specially on the improvement of one specific quartile (called a module) course within the programming line of Creative Technology. The course chosen is the final programming course. It is part of module 6 which takes place in the second year and introduces students to a new programming language (C++) and the implementation of basic algorithms from AI, like breath-first search and Depth-First Search.

The requirements regarding this assignment where complementing the course within education regarding the utilization of libraries. The given education should also fit within the rest of the programming curriculum. Next to that, its course should work as an encouraging method for CreaT'rs to start exploring the possibilities. It should not scare students away from trying out new applications. Currently, the teaching staff experiences that students tend to shy away from these topics. This is quite common within programming (Kay et al., 2000).

For this purpose, a few prototypes regarding possible teaching methods as well as a variety of libraries will be researched, implemented, and evaluated. The implementation takes place as part of this year's curriculum. Afterwards in the evaluation, this study will compare both the difference in teaching methods, the students' perception of each of these methods and the resulting student performance. This research aims at finding a fitting addition to the Creative Technology course Programming for AI as well as ways to incorporate libraries into this course.

In order to do so first and foremost the need for library education within Creative Technology must be researched. The overarching goal is to improve the Creative Technology curriculum hence there is also the need for information regarding the possible libraries. Another key component in library education is the way programming education is conducted and what research regarding this is already available.

Finally, some teaching methods will be implemented and examined more closely. The questions that go hand in hand with this are; How can we best validate this? What teaching methods are currently available? and, what did the implementation of these new teaching methods yield?

CHAPTER 2 – METHOD

Improving a curriculum is no easy task. Due to the changing nature of student bodies, the vast amount of tweakable variables and the interpretations of the researches quite some variability can occur. For instance, students' bodies, classroom settings and personal circumstances. The overall trend within students even shifts with regards to grades, preferred courses and motivation. Next to this, for any specific course last year's drop-outs are likely to retake the course in order to still pass the course. This can cause fluctuations in the earlier established trend of a certain years' performance.

Due to these uncertainties this research has taken a more stable initial exploration. Although the students change completely from one year to the next, teachers can provide a more stable measurement. The teachers are aware of a more general baseline that has been established over the past years and possess a view of what the mission statement of the overall study should be. Unfortunately, this interpretation of the principles regarding the programme are based on their personal opinion and are thus heavily biased by their own opinions.

In order to tackle individual biases, the information assembled from multiple members of the staff should combine into a general front. The qualitative data required will be gathered by interviewing the general staff. This allows for the foundation on which to continue with more explorative research. Having these results also aids the assessment of which topics would or would not be suited for the Creative Technology curriculum. All in All, the input provided by the teachers via the interviews should generate the essential restrictions.

After the essential restrictions have been set a more explorative ideation is required. The three inspirational pillars of this research are: the way programming is taught at Creative Technology, the research available regarding introductory programming courses and the availability of libraries. The latter is even further specified by only regarding libraries available for the OpenFrameworks platform. The information that has been gathered in this stage will provide the specifications to develop a prototype implementation that suits to be evaluated during the programming & AI course.

The realization will be tested during the 2nd quartile of the academic year 2018/2019 and is directly implemented into the curriculum of the students. However, it is only mandatory of the students to perform the original tutorials. For the final examination however, students do need to include a library into their end-assignments, for which the prototype can greatly help to get them on their way.

In order to properly assess the performance of the prototype a quantitative baseline must be established. This baseline will be set utilizing questionnaires that will be conducted during the course. These surveys will test the student populous' motivation for both the general course and the specific component that students are engaged with at that moment. Other questionnaire will test the motivation regarding the prototyped tutorial. In order to test the students motivation the SIMS motivational scale will be used as stated by de Boer & Winnips(2015). The original questionnaire can be found in Appendix E. All surveys are held before students start to work on the end-assignments.

A more qualitative method of exploring the performance of the prototype will be held in parallel. The information will be gathered using panel discussions that allow for more in-depth questions and reflection on the acquired quantitative results. Finally, a closing interview will be held with the staff after the course to round off the result of both the performance of the prototype and the gathered intel.

CHAPTER 3 – STATE OF CREATIVE TECHNOLOGY

The research in this thesis is conducted for and in the context of Creative Technology. As Creative Technology is a rather young study and the title does not fully speak for itself. This leads to the necessity to elaborate on its definition. So, what does the Creative Technology programme entail?

WHAT IS CREATIVE TECHNOLOGY

The modern world is getting more and more complex. The technological possibilities are getting endless while the problems the world copes with today get more and more fuzzy. This requires engineers who are able to define and handle messy problems, analysts who can define many of the possible solutions before they start investigating them and designers that are able to pinpoint why a user is using a device. These new engineers also must be able to identify new problems in an early stage. In this new playing field for professionals it is getting more and more essential to be multidisciplinary and flexible. It requires weird solutions that disrupt the status quo.

“A game played on an interactive LED floor that helps patients convalesce. A wrist band that enables you to track down your friends at a festival. A language test for pre-schoolers using eye tracking. These are all examples of projects carried out by students in our Bachelor’s programme Creative Technology”(Universiteit Twente, 2019)

Creative Technology is a programme that combines the engineering principles of Informatics and Electrical Engineering with the design principles of user experience and persuasive psychology. It aims to mould new engineering students into out-of-the-box thinkers that are eager to make a difference within society. It is an academic study with relatively a lot of hands-on project-based education. And it aims to educate these new engineers.



Figure 1; Creative Technology poster¹

¹ <https://www.utwente.nl/en/education/bachelor/programmes/creative-technology/#more-information>

So Creative Technology is educating the new engineers. Professionals that mix technology with a human touch in order to innovate. Or at least if the Universities own sales pitch is trustworthy. Unfortunately, the text statement above is just that, a sales pitch. So, what is Creative Technology really about?

STATE OF CREATIVE TECHNOLOGY - STAFF INTERVIEWS

In order to establish a general overview of Creative Technology, the teaching and directive staff has been interviewed. Due to time constraints not each member of this community is part of this study. The total amount of interviewed staff members is ten. Among these staff members were mostly module coordinators and directive staff. The few open spots after that had been reserved for less (with Creative Technology) experienced members of staff to balance the scales. The members of staff were all given a very open oral questionnaire regarding their opinion on Creative Technology. It has to be noted that most interviewees were quite startled by the questions received. Luckily, this made the received answers quite diverse and lead to a broader picture of what Creative Technology is and what its alumni should aspire to be.

The interviews have been conducted mostly in Dutch and the questions have gradually introduced during the interview. They ranged for question like “What is a CreaT’r?” to their social need and what, according to them, tools are required in order to be a CreaT’r. The tools required was steered towards technological tools like Android and Python. Due to this research being documented in English the following is only the interpretation of the interview outcomes. The Dutch transcriptions of the interviews can be found in Appendix A.

When asked to describe what a CreaT’r is there is a great diversity in the answers. Although most members of the staff were quite close to each other in overall perception many had a somewhat angled believe or at least give it their own spin. These answers alone are a beautiful symptom of the overall diversity that also exist within the students. Although all staff agrees on the fact that the study combines Informatics, electronics and design, they differ greatly on what they perceive as most valuable. Obviously, favouring their own subject but more interestingly also with the rest of the division.

With this approach Creative Technology tries to educate the engineers of the future. After years of rapid technological innovation most engineering studies have drifted farther and farther apart from each other. In order to pioneer new technological advancements, you have to be an expert with a very small field. This deep dive into a specific topic is what the current educational system is built upon. We have Bachelors that focus on subjects, Masters that focus on specific fields and a doctorate is achieved by those who devote themselves to a specific topic. Creative Technology, on the other hand, focusses on giving students a very broad education. These students need to be the bridges between 'normal' people and the specialist that come out of modern universities. They are able to talk to consumer and retrieve the requirements the engineers need to go to work. They are the pivotal point in our modern society.

In order to be these engineers of the future they need to solve multidisciplinary problems.

In order to solve these multidisciplinary problems, they require a broad skillset that allows them to grasp what is going on within disciplines. But in order to do so, the engineers need to understand these specialists without requiring the specific knowledge. Having this broad outlook on the world enables CreaT'rs to identify fuzzy problems, dream up unorthodox solution and communicate these ideas outwards, with both professionals and the end-users. Recombining existing technology and fashioning something new out of its components.

In order to do so, the need to be quick in following up on the latest technologies. The new engineer will continuously be learning what is new and how he would be able to utilize this. He needs to believe in himself that he is able to prototype with this new technology and uphold a mentality of; "Just do it". The new engineer will be learning throughout all his career and be comfortable with the changes it brings. He needs to own up to his decisions that have brought him his competencies.

All in all, I established a definition describing a CreaT'r as follows;

"An innovator that loves to build creative solutions that are specifically focused on both the client and the intended user. They are comfortable to just jump into new technological advancements and try to use them in interesting prototypes that keep an eye out for humanity and the user. A real provoker that simply can not keep its hands off a fuzzy problem. They do geeky stuff but they are not geeks per se "

But what does the staff believe is missing? Which disruptions in technology should a CreaT'r be invested in? What is the curriculum still missing out of?

In order to answer these practical questions, we also asked the teachers what they thought were the topics a CreaT'r could not miss out on. A question essential to look out for when choosing to do something different. It partly dictates what we should expand on and can give direction in finding which road to go on. For example, in library education there are many different libraries to select from, so which topics would be preferred?

The hottest topic for CreaT'rs was the interpretation of data. Big data is currently big and Creative Technology has always had quite a fondness for visualisations and physicalisations. So, it is not really surprising that this was the most mentioned area for CreaT'rs to dive into.

Being able to prototype on mobile devices is also a close second together with data analytics. In the current curriculum development for the android and iOS platforms is not discussed and some members of staff do feel that this is a shortcoming, certainly as quite a number of graduation projects would benefit from its utilization. Data analysis however is part of the curriculum, it is just not worked out from a computer-run application. And it stands to debate whether students require an explicit course to help them connect the dots.

Other mentions such as IoT, Machine learning, cybersecurity, python and XR have also been mentioned. Machine learning again has a lot in common with data analytics as its main goal is to process large datasets for classification purposes. Cybersecurity is more of a counter to Machine Learning as it tries to block data analytics out with regards to security reasons. Python is an upcoming language that like IoT enables the development of 'smart' products. Finally, XR (the combination of VR and AR) pushes the limits of how we can present users with a different view.

All these mentions are important to keep in mind while searching for appropriate libraries that could potentially be used. Next to these libraries some of the staff members also mentioned other things regarding the development of programming within the curriculum. Informatics is quite a hindrance for many of the diverse students. For them it is essential that they are encouraged and have fun while doing the programming courses within Creative Technology. Otherwise students tend to avoid programming as much as possible. There are students who opt out of doing programming all together during their bachelor assignments. This closed mindset originates in a sort of 'fear' for doing programming. So, it has to be kept in mind that students feel this rejection towards new things within programming when developing a new teaching method.

STATE OF CREATIVE TECHNOLOGY - PROGRAMMING

In order to improve and prototype with the programming education given in Creative Technology, observations have to be made of the current state of the curriculum. These observations are gathered based on previous experience with the four programming courses within Creative technology.

Within the Creative Technology mindset diversity plays a major part. It tries to combine electrical engineering to the business approach of how to sell it. User-centred design is combined to methods for production. Finally, research skills are also part of the fundamentals. The mindset of combining all these fields together is where the need for basic programming stems from. Programming is a means to which a CreaT'r can express himself. The manifestation of his creativity can be found in his coding as a CreaT'r has to make things work.

In order to have working prototypes a CreaT'r needs to be able to program. Hence the need for a certain baseline requirement that all students have to obtain. It however does not end there as there is plenty to gain for those who seek it. The method of teaching rather focusses on the playfulness and spontaneity of the process involved with combining multiple disciplines than structurally correct code. In this, the original idea is valued over perfect execution. The aim is that this results in creative applications and thus favours a tinkering approach.

This tinkering approach is important for the form of the programming education. Lectures are kept relatively short in order for students to have more time to play around with the newly available content. The tutorials that are given can be used as a pathway through the course or simply as steppingstones for creating the end-assignment.

End-assignment are the main measure of skill (within the programming courses at least) and allow students to show off what they have learned and leave enough room for expressing their individual creativity. Utilizing creativity is even a mandatory requirement forcing student to think somewhat out of the box in order to receive a passing grade. This generates a natural divide between students in a good way, it allows for individuality and favours the bold. The end-assignments are the pinnacle of the student performance in programming.

In order to allow students to be brave and explorative they however need quite some guidance along the way. When working out the assignments, questions are meant to arise and challenge the students. This approach does require a lot of individual assistance. For the required assistance Creative technology provides students with a lot of students assistants. Students assistants are elder-years students that are available to answer specific questions and can help out students that encounter issues. These issues can vary from principle misunderstandings to technical difficulties beyond normal expectations. The availability of these student assistance aids the process of going out of the planned curriculum as they can provide a sort of solution manual to common defects and provide specific explanations when concepts have not been grasped by the student naturally.

STATE OF CREATIVE TECHNOLOGY - CONCLUSION

Creative technology cannot simply be described other than a study for those who love to create new things. Due to the great diversity in students and in the fields of interests what is created is quite versatile. The course program tries to educate all these thrifty spirits and shape them into the engineers of the future. By combining software and hardware the study-program tries to teach all students to have a “just do it” mentality. It hopes are on all students diving into emerging topic like the mobile world or the possibilities of big data. Hence the only thing the curriculum fears is closed off student.

The programming education that is given tackles this in the form of ‘fear’ for programming. All students should feel familiar with programming as a tool in which they can express their creative solutions. In order to do so students are encouraged to tinker their way to solutions instead of them executing the process flawlessly. Students come up with their own problem descriptions, creative solutions and final concepts. Along the way the educational program tries to maximally support students in achieving this final concept. In order to do so, Creative Technology requires good introductory programming education.

CHAPTER 4 – STATE OF THE ART ON PROGRAMMING EDUCATION

Within modern engineering studies, the basics of programming has become an essential principle. More and more fields require students to be able to work with, understand and create software. This ever-growing demand asks of the educational system almost everyone possess the ability to write up new algorithms. Yet learning to ‘speak’ computer as a second language has been known to be very difficult due to the lack of time to practise (Cunningham et al., 2016). ‘Speaking’ algorithmic requires a lot of problem-solving skills., which takes a tremendous amount of time and effort to obtain.

HIGH FAILURE RATINGS IN INTRODUCTORY PROGRAMMING COURSES

It is common knowledge to both teachers and students that gaining the ability of programming is no easy task. Introductory programming courses are generally regarded as difficult (Robins, Rountree, & Rountree, 2003). This is reflected by the dropout rates that are associated with these types of studies and courses. The dropout rates of programming courses tend to be high (Robins et al., 2003; Vihavainen, Paksula, & Luukkainen, 2011; Wells, Barry, & Spence, 2012). Furthermore, Kay (2000) describes non-indicative introductory courses as a discouragement to students to continue studying the discipline all together. A trend which is not only perceived by the students who may fail the course. Even the institutions teaching programming courses tend to analyse their ratings carefully. As many institution are judged by the world trough ability to teach programming(Wells et al., 2012). So, there is a great demand for properly taught introductory courses.

In order satisfy the demand for good education an analysis of why the dropout rate is so high is needed. Student drop-out occurs due several reasons. First and foremost, they get discouraged by the complexity of the material(Solomon, 2005). Programming is not a subject that can be learned by remembrance. As with learning to speak a language, it requires the understanding of various concepts of that language combined with the fluency to formulate new combinations with those facets(Cunningham et al., 2016). Language comprehension is needed to make rhyme and reason within a programming structure(Pears et al., 2007; Robins et al., 2003). Robins(2003) adds, that due to the complexity of this task, programming novices experiencing only small improvements during their first introductory programming courses. The students themselves are aware of the steepness of their learning curve resulting in a feeling of being inadequate. Solomon(2005) advocates that, most courses are simply not tailored for complete beginners. A mismatch that makes it highly un motivating to continue solving the

provided problems during tutorials(Vihavainen et al., 2011). Such a motivational low tends to put students off seeking help solving the puzzles presented. And thus, good introductory programming courses should boost the students' motivation.

KEY FACTORS FOR INTRODUCTORY PROGRAMMING COURSES

The high drop-out rates raise the question if normal teaching styles are suited for introductory programming courses. Cunningham et al. (2016) clearly state they are not. Traditional teaching styles tend to focus too much on the theoretical facets such as syntax and styles(Robins et al., 2003). They also do not provide the required amount of practice and hands on experience. This often makes them unnecessarily complex and difficult. Another objection to the architecture of introductory courses in programming is the absence of a storyline throughout the learning experience (Solomon, 2005). The problem that exists however is the diversity of students. According to Wells et al.(2012) there are possibly as many learning styles as there are students and the traditional learning style is generally the worst fit. On the other hand, Pears et al.(2007) surveyed over 170 research papers regarding introductory courses and finds that he cannot conclude what would be the best teaching style. Yet, it is quite evident that the current learning styles have to be improved in order to lower the dropout rates (Pears et al., 2007; Vihavainen et al., 2011). Which results in a clear need for a new method for teaching Introductory programming.

A couple of key factors for learning programming have already been defined. As mentioned above, Ability for students to practice and the teaching strategy used are significant key factors in the teaching of introductory programming. However, those are not the only two factors that must be considered. Fluency with the language components is also a key factor that can be specifically trained. Solomon (2005) speaks of the abstraction level in being able to translate code into normal language. This is also brought forward by Pears(Pears et al., 2007) who also advocates using metaphor and paradigms for creating tangible explanations. These two factors both aid the development of fluency in a language. Students should be taught why an algorithm is designed in a specific way and what makes it effective. It is important that we teach the correct structures with which code is written (Robins et al., 2003). The third key learning factor for programming is gained competence of the structures used within a course.

But those are not the only key factors. Robins (2003) also explains the importance of exploration within our programming education. He states the added value of being able to think with programming. Kirschner(2013) even states that experiencing the creative process of programming adds significantly to the learning experience. The tinkering approach is characterized by being playful and experimental(Mader & Dertien, 2016; Robins et al., 2003; Wells et al., 2012). And although these elements are often academically questioning, they are of undoubtable value for every programming courses. Mader and Dertien(2016) states that it teaches not only the ability to pick up new technologies, it also trains the observation and reflection skills. Tinkering, thus, is of significant value to especially academic programmes.

Besides the programming specific key factors, the general key factors identified for learning should also be considered. Programming language are called a language for a reason. As with other languages, it is very important in programming that the used mental models, grammar and good practises are expressed in their correct forms(Pears et al., 2007; Robins et al., 2003). Students should not only learn to understand a programming language, they should also be able to utilize that language own syntax in building their own programs.

Finally, students strongly perceive their own ability to succeed as a key factor. Students also perceived they require a lot of time to practise in order to master programming (Cunningham et al., 2016; Hawi, 2010). It is important that a course provides the possibility to practise as it raises the students' perception of their own abilities. This is partly due to the nature of solving problems within a programming assignment. It is known that solving information problems is major cognitive endeavour for most students (Brand-Gruwel, Wopereis, & Walraven, 2009; Kirschner & van Merriënboer, 2013). And although this is seen as one of the core 21st century skills (Anderman, Sinatra, & Gray, 2012; Voogt & Roblin, 2010), doing so within the context of programming can be daunting and close to insoluble. Here the comparison with a second language can be made again. Solving erroneous code within self-written programming exercises is comparable to finding the correct usage of a verb in a foreign language. If the student has no connections with that language it is close to impossible to find. Vihavainen (2011) suggests that this can be countered by the availability of continues feedback provided by teaching assistants. These more experienced students can be a translator for students in need of extra guidance. All in all, students require enough opportunities to develop their programming skills in both time and assistance.

The pressure as explained in the first chapter is especially felt by students how tend to fear programming. This raises the need for a good method for teaching Introductory programming. Hence, the material given should not be too complex and students should be given enough time to practise with it. Next to that the material should be boosting their confidence and enhance the students' ability to be fluent within the language and its syntax. During this process students require a lot of assistance in order to keep up the motivation. This makes Tinkering during introductory programming courses especially useful for Creative Technology, like Mader (2018) has mentioned.

TEACHING METHODS FOR INTRODUCTORY PROGRAMMING COURSES

So, what would be suitable teaching methods for programming courses but first let us set a definition for what a teaching method is. A necessity due to the fact that not everybody fully agrees on the meaning of teaching method and for the rest of this research it is important that a clear definition is chosen. For some a teaching method is the materially based set of books and instructions upon which the educational program is built. This can for instance be seen in educational publishers that utilize the same structure for teaching all kinds of languages(Malmberg). For others teaching methods describe the utilized strategy and form from which the education takes its shape (Teach.com). In this paper the term Teaching Method refers to the way education is brought upon students; how education is tutored. In this context the pedagogical style and management strategies used for classroom instruction are mainly focused on.

A teaching method is highly personal and even within the same general strategy each and every teacher has a somewhat altered approach. The eventual utilization of educational methods is influenced by the teachers chosen strategy, the given subject, classroom demographic and the intended goal. Hence, what works for mathematics is not necessarily a good fit for programming and what works for programming in another study program is not necessarily a good fit for Creative Technology.

The variety of Teaching strategies can however be classified. There are two major parameters which can be used to make a distinction between strategies. First of all, a strategy can be teacher-centred or student-centred approach. The oldest form of institutional education is heavily teacher-centred, with the utilization of lectures, instructions and assessments. It perceives students as empty vessels that have to be filled with knowledge as efficiently as possible. Student-centred learning on the other hand gives students a role that is a lot more pro-active. A teacher will try to stimulate and facilitate students but without forcing a student to take a prescribed path. This provides the students with ownership over their own learning experience. In the end the centeredness of education is mainly based on who has the most dominant role within the educational strategy.

Secondly it can be dependent on high-tech material or rather utilize low-tech material to support the strategy. When talking about gamification, laptop-utilization or electronic learning environments the chosen path is utilizing a lot of high-tech features. Mylabs-plus, Jupyter and Spinoza are good examples of high-tech features that enable students in their learning. Low-tech solutions however show great benefits for memorization and allow for a lot more interaction. Having hands on experience may benefit students greatly and allows for a much more open setting. This does however sprout the question if programming can ever be thought in a low-tech setting, as it will always require computer-systems to be programmed. In order to still make a somewhat closer comparison we have to adept this second parameter to whether the teaching style utilized more or less tactile experiences as support.

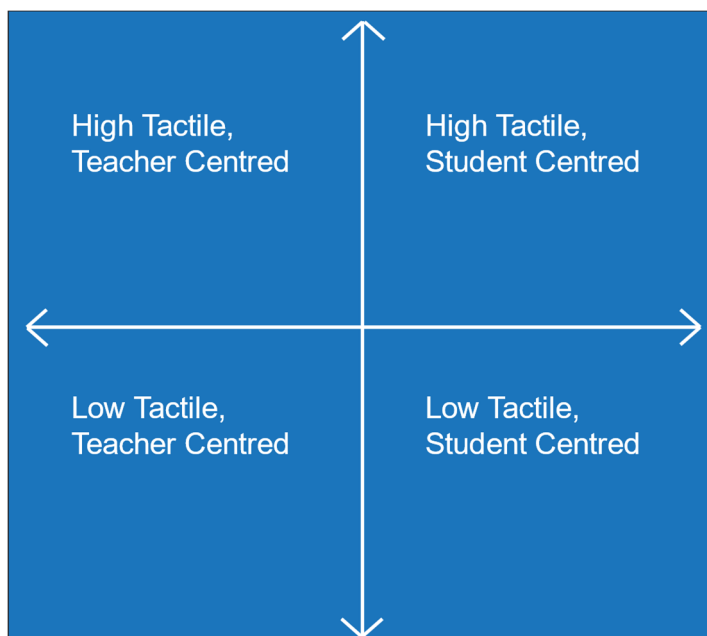


Figure 3; Map of the dimensions of teaching strategies

EXAMPLES OF TEACHING METHODS

A good example of a very tactile educational method is Greenfoot(Kölling, 2010) where normal raw java based coding is complemented with a premade coding structure and supplied visuals. This enables the student to try programming for the first time but with the correct structure they need to allow them to have an overall connectiveness.

Processing, on the other hand does not supply this structure but supplies students with shortcuts to make it easier to build this structure for themselves. Without the utilization of header files, for example, it is a lot easier to make cross-references between objects. Processing also supplies easier function calls where native java would require reference variables and much more includes.

STUDENT ENGAGEMENT

These days, students are expected to engage with the real world more and more. It is important to analyse what happens in different life spheres. These cross-comparisons are valued for their ability to lead to new insights by not only focussing on the theoretical but also the practical implication when applied across fields. A good way to do so is by making witness how the knowledge that they were learning applied to the real world. This also includes giving students exercises that have no right or wrong answer. Simply working on something real generates engagement and it allows for better intrinsic motivation and subsequently better grades. The idea is to get students engaged and to connect their learning to the real world. If teachers can show them how what they are teaching connects to the real world then their own brain cells are going to connect them and associate them. Hence, engagement can play a major role in raising the students level of understanding and so their grades(Conley & French, 2014).

Jackson (2012) presents that state that utilizing the engagement principle resulted in 91% of students achieving passing grades. As new methods of teaching should try to improve the quality of education, involving students with physical scenarios improves the education significantly(Kuh, 2003).

STUDENTS OWNERSHIP

Another way to improve student achievement is through supporting student ownership of learning. Student ownership is teaching students to play an active role within their learning as it benefits their performance in the long run. By promoting student goal setting, self-assessment, and self-determination students become meaningfully engaged in their learning(Chan, Graham-Day, Ressa, Peters, & Konrad, 2014). They gain a better understanding of the importance of learning targets, ways to collect and document evidence of their skills, and how to evaluate themselves. They also have to clarify additional learning needs in order to keep them working on the goal of improving themselves.

The Flipped Classroom

All bachelor programme of the university of Twente are organised as thematic modules with a project and several courses. This structure is defined by the Twents Educational Model (T. Universiteit Twente, 2013) and tends to emphasis the practical implications of the world in the theoretical background. A form that neatly plays to the principles of ownership and engagement. In doing so it already aims at making education tactile, but it does keep a teacher centred approach. By applying a 'just-too-late' teaching model, meaning knowledge is offered to students a after the initial encounter, students will first try to come up with their own solutions.

However, there is another option to do so. In making information readily available to students all the time and not discussing the material in lectures, students have to look the initial information up themselves. Not supplying lectures creates space to discuss subject matter and requires active participation in the form of formulating questions and asking for help. This slightly controversial method of teaching is called Flipping the Classroom (Bishop & Verleger, 2013).

The problems regarding Flipped Classroom are however that it seems to have varying results. In primary education this a very booming topic while it is utilization in higher education and universities is rather provocative. Here at the University of Twente a previous study noted that the pass-rate of a course went from 80% to only 66% and that participation dropped dramatically (Gommer, Hermsen, & Zwier, 2016). This reflects on a lack of ownership as students were reported they only started studying at the last moment as there was no stimulant to put in efforts earlier. It can be debated why students struggling with the concept of such a flipped classroom. Most students like the availability of the content but the Flipped classroom setting is appreciated much less (de Boer & Winnips, 2015). Flipped classroom tends to favour the pro-active students while the more passive students view it as a reason to up off their work until the very last moment. Furthermore, it is very hard to create a structured participation without the utilization of deadlines or mandatory exercises.

CHAPTER 5 – STATE OF THE ART ON LIBRARIES (FOR OPENFRAMEWORKS AND PROCESSING)

The programming course in Module 6 is quite different from the other 3 programming courses with regards to the language utilized. After having programmed in Processing for the entirety of the first year, this second years course is given in C++. One of the reasons for this is to show students that their fluency in Processing is also valuable when working with a different language. Most students regard this as quite difficult in the beginning, but it offers them confidence in their own ability in the long run. This serves the goal of making students capable and comfortable with the challenges that being a modern engineer will bring them. In order to still provide students with some familiarity OpenFrameworks is introduced as both toolkits are quite similar. Next to that OpenFrameworks is able to process much more data without generating lag² allowing for 'larger' applications to be build and run. This allows students to get to work with C++ rather quickly.

Another reason is the possibilities that are offered by the OpenFrameworks platform in regard to libraries. Open frameworks has over 1000 different libraries (addons as they are called) that are developed in open source environments and allow the use of DLR camera's and Dlib machine learning algorithms³. The downside of this is that many of these algorithms are not very well validated nor documented. Luckily, OpenFrameworks itself came out with a list of includes libraries. These libraries are the most commonly used and thus quite stable. The so-called OFxAddons are preinstalled and can be included using the application generator provided by OpenFrameworks. There are official 14 OFxAddons within the current OpenFrameworks version 0.10.0.

Next to the official libraries, this research also looked into a few other facets of OpenFrameworks and similar libraries for processing. Although the course is by default given in C++, students have the possibility to opt for keeping java (via Processing) as the used language. Hence, the module 6 course is also in need of some library education for the processing platform. As this is not the main goal of this research only some of the libraries that are quite similar to the once that are part of the OFxAddons are presented. Secondly, the integral part of the OpenFrameworks platform that is called openGL is also mentioned as it does closely resemble a library due to the direct instructions in runs on the GPU.

² <https://www.youtube.com/watch?v=NZG3g0NRR4I>

³ <http://ofxaddons.com/>

GRAPHICS

Graphics are an important part of the current programming line utilized for the Creative Technology curriculum. This is contrary to most other introductory programming courses in which a more mathematical approach is taken. This gives Creative Technology students an advantage regarding the use of visuals and interaction. Utilizing graphical libraries within the Creative Technology curriculum enables students to expand the already available ability to create graphical interfaces. Within OpenFrameworks, there are libraries available that could be of great use to Creat'rs.

OFxSVG

SVG is a standard type for images saved as scalable vectors (scalable vector graphic). This format also makes it a lot easier to change small bits of the metadata in order to create interactions or animations⁴. OF has a simple library allowing easy usage and implementations for the SVG technique.

OFxAssimpModelLoader

Within the curriculum students have the freedom to explore 3D models and animation. Some students even chose to extend their given base-level knowledge with the advanced courses offered in the 5th module of the curriculum. OFxAssimpModelLoader is a library that allows the importation of 3D models so they can be utilized within an application⁵. This could assist students in building 3D applications or by creating interactivity within their 3D models.

OpenGL

OpenGL is not truly a library. Officially OpenGL is simply utilizing the commands pre-programmed on GPU's. Not all GPU manufactures pre-program the same commands into their GPUs. Due to this reason both programming languages have libraries to allow for easy compatibility.

⁴ <https://www.linkedin.com/learning/illustrator-cc-for-web-design-svg/what-is-svg-and-why-should-you-use-it>

⁵ <https://www.youtube.com/watch?v=V1EwCHHxFUs>

OpenGL for Processing

Within processing openGL is already used behind the screens. This regards saving images as textures (a more common open GL format) allowing for the images to be processed on the GPU instead of the CPU. This, for instance, allows processing to handle 10 images of circles quicker than 10 CPU drawn image vectors. However, utilizing direct openGL within processing on a structured basis can lead to major problems⁶. This is mainly as communication regarding code will become confusing. It is such an alteration to the normal code structure that many peers will become completely disoriented to the point that code becomes comprehensible. Apart from that, it is obviously allowed to use direct openGL commands within processing.

OFxGL

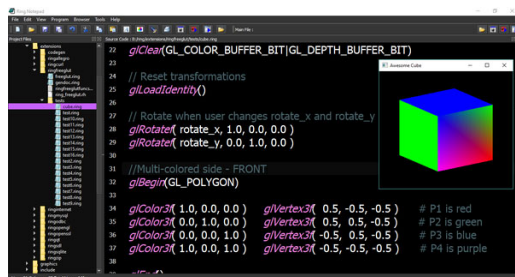


Figure 4; basic OpenGL vecor implementaton⁷

OpenGL can easily be utilized within OpenFrameworks. A wrapper exists that even helps with the use of openGL. It is still possible to hardcode openGL as would normally be the case within C++⁷. On the other hand, the wrapper allows to quickly and easily call OpenGL commands. This is also a lot more efficient as it uses internal references allowing for faster shader and texture placement.

OFxGui

The utilization of OFxGui allows students to build more advanced graphical user interfaces (GUI's). It not only provides the user with more functional and better-looking buttons, it also supplies active variables (variables dependent on the user). These active variables are updated by dedicated event-handlers that allow swifter user-interaction.

⁶ <https://github.com/processing/processing/wiki/Advanced-OpenGL>

⁷ <https://www.quora.com/What-language-is-OpenGL-code-written-in>

MOBILE DEVICES

Many modern technologies make use of the abundant availability of mobile devices. For a CreaT'r to start using mobile devices as controllers, actuators or completely run their applications these libraries are interesting. In the survey of staff this was the single most mentioned competence that was believed to add significantly to the curriculum.

OFxAccelerometer

OFxAccelerometer is one of the easiest ways to use mobile phones as controllers. It allows to use the sensors embedded within the phone as actuators for running applications. Unfortunately, it is quite a closed environment and only allows the use of the accelerometer (hence the name). The library has never been further developed as a library but is the basis of many other libraries that allow embedded sensor usage.

ANDROID

Android is 'the' most commonly used mobile operating system (Melanson, 2013). It is completely open source and maintained by Google. As 90% of the mobile devices allow Android applications to run it is one of the best methods to allow applications to be controlled or run by mobile devices.

OFxAndroid

The OFxAndroid library allows users to build standalone applications for Android in the C++ language. It does so by compiling the OpenFrameworks layout into android while keeping the normal functions and introducing functions like (physical) button handlers. It also allows you to build applications that utilize functions like the GPS or Network protocols that are exclusive to phones.

Android for Processing

The Processing platform started off as an easy to use toolbox for Java with its own editor.

However, its success in being so low key and the visual focus have driven processing a lot further. Processing in itself is officially a library. This library is now also available for android allowing code that is written in the processing language to be run on android devices. This leads to a very intuitive and plug and play approach for writing android applications in the processing language.

IOS

iOS is Apple's alternative to android and the operating system that runs on all Apple devices. It is not open source and thus quite a bit harder to get up and running. Being a developer for Apple is regarded as quite an esteemed skill. Unfortunately, working with iOS devices can be quite a hassle as it comes with a lot of regulation for the developer.

OFxIOS

It is possible to run home made OF applications on an iOS device, however there are quite some requisites. Firstly, you will have to use Xcode as development studio⁸. Next to that you have to be a registered Apple developer which cost \$99 dollar. Due to Apples frequent updates it is not 100% certain your current version of iOS is still compatible. All in All, using iOS as a mobile platform for OF is quite the hassle.

IOS for Processing

Unfortunately, iOS compatibility for processing does not exist

WEB / NETWORK

OFxEmscripten

C++ executables are not able to run on the internet. By using OFxEmscripten it however is possible to run OpenFrameworks applications on websites⁹. It compiles the generated C++ code into JavaScript which in turn can be run embedded into a HTML page. Enabling the developer to reach a wider audience with his application.

⁸ <https://openframeworks.cc/>

⁹ <https://www.youtube.com/watch?v=QxRAj-EzVZA>

OFxXmlSettings

Data, especially online data, is mostly stored as XML¹⁰. Databases, for instance, are able to deliver the data in this format and utilizing this in OpenFrameworks allows the user to make applications that store or retrieve data from an online database. The OFxXmlSettings allows easy connections and interpretations of this data-format for the user to work with.

OFxNetwork

A library that allows for TCP and UDP communications. Effectively connection an application to the internet and thus allowing for online database connections and remote communication.

OFxOsc (Open Sound Control)

OSC is an easy network protocol for sending the most common data types. Osc is an open standard replacement for MIDI. This means, with a requisition that the correct hardware is used, that you can control your application advanced pieces of hardware. To name a few; x- OSC (high performance Arduino's), large XLR transceivers and MIDI controllers like APC40 mkII or an Ipad.



Figure 5; Open Sound Control controller¹¹

¹⁰ <https://www.youtube.com/watch?v=Vkn3VWVUcX0>

¹¹ <http://www.akaipro.com/products/pad-controllers/apc-40-mkii>

OFxPoco

Poco is a collection of additions to C++ in order to get it up the same standard as Java and other more modern languages¹². This was mostly regarding Network communication. In the 10 years it has been developed it became much more and claims to be the only library you would need for communication and networking.

VISION

When linking the real world into application it often occurs that the easiest way to do so is with cameras. Many libraries that exist for OpenFrameworks provide solutions to link visual devices like webcams and DLR cameras to applications. The most commonly used are Kinect and OpenCV;

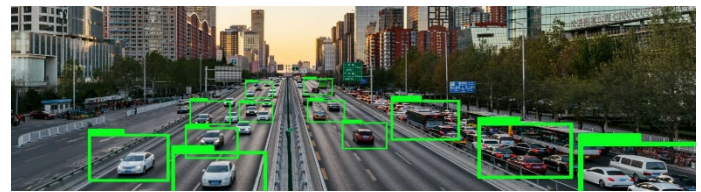


Figure 6; Example of a Computer vision application¹³

OFxKinect

Kinect is a readily available platform that allows more than normal imagery. It combines RGB images with infrared and sound. This allows for simple 3D positioning as well as normal VGA pictures and sound capturing. There are a lot of examples available and it is really easy to start using the Kinect, however it has been discontinued, and thus obtaining one is getting harder. Another downside of the Kinect is the framerate.

OFxOpenCV

OpenCV is especially designed to analyse images and is one of the libraries on which OFxKinect is built. It is somewhat sturdier and contains a lot more functions that allow for fast image processing. This makes it somewhat harder to get working but allows for a lot more functionalities.

¹² <https://www.youtube.com/watch?v=riw7sQ61gSU>

¹³ <https://www.surrey.ac.uk/postgraduate/computer-vision-robotics-and-machine-learning-msc-2019>

This page was intentionally left blank

CHAPTER 6 – REALIZATION

SPECIFICATION

During the ideation-phase extensive inquiry was done in what the requirements should be addressed by the prototype tutorials. As the tutorials would directly be tested on students whose study-career was depending on it, one of the important requirements was the non-intrusiveness of the experiment. Obviously, students were still scheduled to attend the tutorials, however their participation should not affect their results directly. It would be absolutely fine if they gained advantage from the experience, but attendance was not mandatory. In order to test their utilisation of the tutorial, the use of a library was included in the end-assignment. The students were allowed to use any library to complete this requirement.

A requirement was also set on the easier use of non-taught libraries. As CreaT'rs should be able to quickly adapt to use new technologies, their ability to use their understanding of libraries on new matter should be stimulated as well. This again strengthens the position of the requirement in which they are allowed to use other libraries to show of the newly gained skill with libraries.

The difference between novice and the more experienced programmers within the student-body is also a division that has to be taken into account. This generally leads to the possibility to express more or less effort in their end-assignments and the consequent grades they get for these assignments. It has to be considered that the divide between good and bad programmers has been growing since the first module (in which most students touch on programming for the first time). This divide is acknowledged by the curriculum and the more novice programmers are offered a lower bar for their assessment. By providing understandable tutorials however we hope all students have an equal change at least to grasp the fundamental principles regarding the subject matter.

In order to further support the novice programmers, the prototype requires multiple levels of difficulty. This is a principle utilized by Creative Technology to allow more experienced students to directly aim for higher grades. In the 3rd programming course the difficulty level is simply set as on a 6, 8, 10 scale indicating the level students are aiming at. Some students know that they are novices and stick with level 6 while the more developed programmers have a clear perspective of what to do in order to aim for a 10. Interestingly enough certainly level 8 helps students to strive to have the certainty of getting a passing grade. The well-motivated novice students try to aim for this level, with the knowledge they will probably get some reduction, in order to be absolutely sure of getting a passing grade.

Most requirements have been made in order to minimize the impact of the research. Students should never have to deal with implications due to errors that occur within the research. From this also stems the absolute anonymity of the questionnaires and the voluntary participation. If anything were to happen with the research, the course planning should easily be able to adapt and accommodate students in still passing the course.

IMPLEMENTATION OF THE TUTORIALS INTO THE 2018 COURSE

One of the requirements is that the evaluation had to be fitted to running course planning of creative technology. The course normally had a week by week planning. The planning consisted of a division for all the topics that were covered during the course. The problem with this previous planning was mainly its lack of timeslots to implement the tutorial. In order to create the space needed, the schedule was altered lowering the requirements for the already existing tutorials. The change in study load resulted in a 2-week period dedicated to the new library education prototype. The 2-week period was filled with 2 new tutorials. As the prototype could not be too intrusive to the existing program, the use of the tutorial material was not mandatory.

The scoring system for the course was largely kept in place. 20% of the course was based on signing off the first 2 tutorials. In this, the prototype was not included in order to keep the intrusiveness to a minimal. As these practical assignments were meant to be done in pairs, a written exam measuring up to 20% of the student's grade was also included. Finally, all students had to close the course with an end-assignment which they had to present during an oral examination. In order to measure the use of libraries and the effectiveness of the tutorials, use of a library was mandatory within this assignment together with physical interaction and the utilization of a course-taught algorithm. The utilization of a library could later be dropped to minimize the intrusiveness. All these assessments combined resulted in the student's final grade.

This resulted in a times-schedule (Figure 7) which allowed for intermittent testing. After the students finished with the maze search algorithms, they will be asked to fill in the first questionnaire. This questionnaire will be held to establish a baseline indication regarding their motivation for both the programming course in general and specifically for utilizing maze search algorithms.

A second and third questionnaire will be conducted in week 6 after students have finished both subsequent library tutorials. These are specifically held to evaluate the motivation regarding the new tutorials. In the end, the data collected by all questionnaires will provide the qualitative data required to take conclusions regarding student motivation during the library tutorials.

Besides quantitative data qualitative data is also collected. Due to the small scale of the implementation in this thesis this data will hopefully provide a more insightful overview of the effect the implemented prototype has on the curriculum. The qualitative data is gathered using panel interviews conducted with the student assistants, the students themselves and the teaching staff in that order. Directly after the tutorials in week 6 all teaching assistants will respond on their perception of how the course went so far and specifically the prototype tutorials. The students themselves will be interviewed on their interpretation of the survey results and their general opinion of the course. This panel interview will be held just before the oral exams in order for their performance to not have effect on the results. Finally, after the term has ended the teachers will get together to evaluate the course, the impact of the prototype and if the desired effect has been obtained.

<i>Week</i>	<i>Theme</i>	<i>Evaluation moments</i>
1	Introduction into OpenFrameworks	
2	Maze Search Algorithms	
3	TicTacToe	Student Questionnaire 1
4	TicTacToe	
5	OpenCV	
6	OpenCV + OFxGui	Student Questionnaire 2 & 3
7	End-assignments	Panel interview with teaching assistants
8	End-assignments	Panel interview with students
9	Oral Exams	
10	Resits	Panel interview with teaching staff

Figure 7; Course schedule by week

PROTOTYPED COMPONENT

Prototyping for educational models can be quite tricky. Firstly, because quite an extensive user-group is needed in order to get useful results. Secondly because all tests have to be planned beforehand. Having an evaluation-round within a curriculum means the prototype has to fit and the required timeslots have to be reserved. Another cycle of adaptation and re-evaluations has to wait for next time the course is given as students cannot really learn the same thing twice. In order to still use a somewhat tinkering design method the original principles were unaltered while finetuning other variables was used to improve the overall quality of the prototype. These alterations were evaluated by the teaching assistants and combined led to the improvements. It has to be kept in mind that the level of the teaching assistants is a lot higher than that of the students. It might have caused errors in the original design to slip through that will be encountered during the student evaluation.

The prototyped courses will borrow heavily from the Flipped Classroom methodology. This is partly due to the already existing nature of the tutorials given in Creative Technology, in which students have to tinker their way through exercises. For this they, however, do not get a lecture to introduce them. The study material will be available before the students have a need for it so they can use the system as intended.

THE ADDED TUTORIALS

In total, 2 tutorials will be developed and evaluated. These tutorials are evaluated during the Programming for AI course in 2018-2019 in conjunction with the tutorials that have been part of the course for the last 2 years. For the first tutorial (Appendix C) a conventional way of the flipped classroom has been implemented. The tutorial was completely based on the OFxGui library and the Flipped classroom principle. The OFxGui library is available as a standard within the OpenFrameworks platform. Students were supplied with a list of YouTube video's that could guide them through utilizing OFxGui. In accordance with the Flipped classroom principles no lectures were given. Students, however, did have the possibility to ask questions regarding the tutorial exercises.

The second tutorial (Appendix D) will utilize a newly designed teaching method. This new form of teaching is heavily inspired by the flipped classroom model. The method draws from the same principles of ownership and preparations before going to the classes. It however does not set students completely free from the more traditional forms of lectures and exercises. This method, named the Guided Read-in, consist of a strong tutorial assignment. The tutorial contains a readable introduction (the Read-in), a somewhat more advanced Q&A lecture and guiding tutorial questions.

The readable lecture allows students to, at their own pace, prepare by understanding the fundamental principles of the study material. The Read-in provides an example that is broken up bit by bit to allow students to understand what is expected of them and to show which steps are necessary to solve the problem. Certainly, with programming education this could help to get a feel for the structure and mechanisms used in the example.

After students have had the time to go through the material, they are set to perform tutorial exercises. These exercises are set up in order for the students to find out where they will strike difficulties within the process. This allows the students to start off at their own pace and ask for help only when they need it.

Whenever the teacher develops the feeling that students are struggling or there are more advanced principles to be explained he can intervene. By giving a small lecture he is able to directly target the students' need for extra explanation. Lectures could cover the understanding of enhanced principles or simply on the frequently asked questions. This allows students to quickly continue with the exercises.

In the end, students will have to implement the new principles themselves into their end-assignments. It is expected of students to fulfil the mandatory requirement for using a library. Students that will keep up with their work and find out the requirement early have plenty of opportunity to get any library working. The laggards, however, will probably utilize OFxGui as it is really easy to catch up with the content due to the availability of online lectures.

The availability of this material can also be beneficial for students outside of the course as it can serve as a crash-course into the topic. This is especially valuable for a program like Creative Technology as it allows other students to use earlier gained knowledge and the new material in their projects or courses.

Utilizing the method also hopes to make the gap between students smaller as it teaches students from approximately the same level. This does require students to do the for them appropriate preparation. A process which requires more novice programmers to do quite some preparation but helps them to not start off completely blank at the beginning of the subject.

In accordance with the time available and the two described prototypes, a time schedule was made corresponding to the available hours per week. This schedule can be seen in Figure 8 and also includes time to sign off the tutorial, however this will not be necessary, as there are no mandatory sign offs included in the prototype during the evaluation.

	<i>Normal</i>	<i>Guided Read in</i>	<i>Flipped Classroom</i>
	<i>Search algorithms</i>	<i>OpenCV</i>	<i>OFxGui</i>
<i>1h</i>	<i>Lecture</i>	<i>Hand out example</i>	<i>Hand out tutorial/videos¹⁴</i>
<i>2h</i>	<i>Tutorial</i>	<i>Tutorial</i>	<i>Tutorial</i>
<i>3h</i>	<i>Tutorial</i>	<i>Lecture</i>	<i>Tutorial</i>
<i>4h – 7h</i>	<i>Tutorial</i>	<i>Tutorial</i>	<i>Tutorial</i>
<i>8h</i>	<i>Sign off</i>	<i>Sign off</i>	<i>Sign off</i>

Figure 8; Timeschedule per tutorial

EVALUATION RUBRIC

In order to evaluate the student during the altered end-assignment an evaluation rubric (Appendix B) was created which allowed for the previously known 6, 8, 10-level scale. In consultation with the teaching staff the correct implementation of the OpenCV library had been set to the 8-level, together with the decently complicated applications and proper event handling on the part of the physicalization. The less significant or simpler libraries were valued on a 6 alongside simple Arduino connections assignment that still look like the given example code. For the 10-level no requirements were set. This is done to have the examiners decide what was worthy of a nine or ten. Examiner, however, got the freedom to evaluate any assignment disregards the rubric of the complexness of the assignment was in line with any of the other levels.

¹⁴ could have been done even earlier than the start of the course.

CHAPTER 7 – IMPLEMENTATION

EXPLORATION OF THE LIBRARIES

The list of libraries generated in the state of the art has been used to further explore the possibilities of utilization within Creative Technology. In order to do so efficiently this list has been ranked by usefulness and potential. In this order the potential libraries have shortly been researched and evaluated as potential libraries for the course.

First of all, Kinect was tested due to its prior engagement within the course. The Kinect library for OpenFrameworks has already been used by some students in the past. Unfortunately, time, and updates, have made it increasingly difficult to connect and run the Kinect on windows devices. Most of the drivers are not compatible with the base system anymore. This does not allow for a smooth connection and requires the students to dig deep into the workings of drivers and manually implement custom drivers. Certainly, making manual customization has had a big effect on the evaluation of the Kinect and has thus not been chosen as one of the libraries utilized in the final prototypes.

Another such example has been the Android development possibilities within OpenFrameworks. Due to the extensive installations of SDKs getting OFxAndroid to run for the first time is a very painstakingly long process which takes longer than a normal tutorial session. Next to that, it requires a specific programming environment (namely Xcode). As most of the students within the course are programming in Visual Studio, the environment the course also sets as standard for windows users, changing this to Xcode for a single tutorial is not advisable. The Macintosh users, however, could try to utilize Android but this is certainly a minority and it has not been possible to test this further.

OpenGL helped to identify another branch of undesirable behaviour in a library designed for educational purposes. Although the possibilities of OpenGL are nearly endless and getting it to run on a device requires only that the library is allocated when generating the new project, the concepts behind it are quite complex. In order to utilize the benefits of implementing OpenGL, students need to grapple with very difficult processes and design methods for every simple component. One of the reasons platforms like OpenFrameworks are used in education is because it allows the student to very quickly generate visual output, utilizing OpenGL would imply that drawing a rectangle, a thing which students learn in their first week of programming. Now becomes quite an extensive process for which they have to write the method themselves. In order for a library to be appropriate for teaching purposes the handled concept should be simple.

One of the utilized libraries has been OpenCV. Computer vision is an expanding field that allows for a lot of versatility and new opportunities. Next to this, getting started with OpenCV requires very little set up time and requires simple hardware such as a webcam (integrated in most students' laptops). Computer vision is the art of trying to make sense of images which makes working with very tactile and will always generate images. OpenCV is visually very strong and rewarding to work with. Allowing students to pick up on this trend does however require basic knowledge regarding the principles of digital imagery. Although the concepts are quite difficult to master, the basics are explainable and allow for quite a steep learning curve. It can be quite a challenge for most students, but this also creates satisfaction for those who get it to work. Finally, OpenCV can be a good introduction for using Kinect as well, making the education versatile and broadly applicable. OpenCV was chosen as one of the utilized libraries because it brings a broad range of possibilities for the students willing to go the extra mile.

The Final selected library that is implemented in the experiment has been OFxGui. OFxGui is very easy to use and has a very nice and short learning curve. Having such an understandable library also allows the students that require some more assistance to utilize features they might like without having to program the functions within OFxGui themselves. These functions in their own turn create a lot of opportunities in using variations within programs. The more hardcoded variables in code can, with the use of OFxGui, easily be tweaked. A quite interesting utilization of this feature is the calibrating settings for computer vision. Hence, the combination of both the utilized libraries can be worth the pursuit. It is this complete versatility and the short learning curve that make OFxGui one of the pursued libraries.

CHAPTER 8 – RESULTS

STUDENTS

The students view regarding educational models is very important. They may not be an authority regarding good educational methods (Kirschner & van Merriënboer, 2013) but student in the end are the users. In order to acquire an overall view student have been surveyed regarding their motivation towards the tutorials. Next to this, a panel interview has been held with the students to enquire qualitative data regarding the tutorials and interpretations.

QUESTIONNAIRES

After conducting both questionnaires (Appendix F & G) the data has been evaluated for its correctness. Factor analyses has proven that all categories have significant correlation for investigating research (Nunnally, 1978). Question 10 however did not show to be properly correlated. Question 10 is about the student's personal decision to follow the programming for AI course and likes it to internal regulatory motivation. Student on the other hand are not voluntarily enrolled in the course. This makes the question somewhat ambiguous towards why the students would choose a specific score. As this might well be the reason for the overall lower score. It was decided that Question 10 was dropped all together which subsequently boosted the α coefficients of the questionnaire.

The statistical scores reported by both show that the intrinsic motivation of students on average produces a 4 out of 7 with an $\alpha > .90$, reporting that Creative Technology students are on average at least partly motivated to study programming. The Internal Regulatory results however show that this is partly due to their believe in the value it adds to their abilities (4.5 out of 7, $\alpha > .80$). The largest contributor to their motivation however is External Regulation. As the highest scoring impact factor, with 5.5 out of 7, $\alpha > .90$, shows us that students feel strongly that they have to follow the course as the program requires them to do so. As the course is part of the mandatory programme of Creative Technology this is as expected. Luckily, students are not completely unmotivated for the course. However, the questionnaire did show a significant decrease between the first and second questionnaire. Originally, students were very neutral about their amotivation towards the subject scoring 3.4 out of 7, $\alpha > .80$, however this grew significantly, 4.6 out of 7, $\alpha > .75$, at the time of the second questionnaire.

More details regarding the statistical analysis can be found in Appendix J

This however does imply that students were less motivated for the prototype tutorial than they were for the reference tutorial, certainly as the other scores also tended to sway toward a more negative result for the new course. It however has to be noted as well that this was also reflected by the attendance of the students. Most students stopped to show up after the initial mandatory components had finished and hence did the number of students filling in the questionnaire dropped from 61 to 28 entries.

EVALUATION SESSION

In conjunction with the conducted questionnaire a student panel was formed in order to obtain more qualitative data regarding the prototype. These students were selected by their own voluntary sign up. The interview was conducted just before students would participate in their oral exams in order for their grade not to affect their opinion of the course. As the interview was conducted after the results of the questionnaire had come in, the panel was also asked for their interpretation of the questionnaires results.

First some general feedback towards the course came to light. In this, students liked the overall concept of the course. The course was very well integrated with the rest of the module. This helped in getting an overview of what they were taught. In this de OpenCV Tutorial fitted somewhat uncanny as they could directly link it to rest of the course. Students also missed more practical examples of why they would use the things taught in the course. In this the new tutorials were a lot stronger, they clearly showed what they could be used for. It applied more strongly to the OpenCV tutorial than it did to the OFxGui tutorial.

Another thing the students mentioned was their struggles with a new programming toolbox. OpenFrameworks did grow on the students but they clearly mentioned that being forced into this new toolbox was quite a struggle. A solution provided by the students was making the library part optional. This would allow the students that got completely stuck with OpenFrameworks to focus on either the library or the toolbox.

Thirdly students mentioned that strict schedules helped them in evaluating what had priority and what they should focus on. For them it helped to know what they should be working on in order for them to focus on the task at hand. Paired with this came their request for a programming week (preferably just before the oral exams).

In the end they liked the extra tutorials offered regarding libraries so much they actually asked for more likewise tutorials. The supply of possibilities helps them to keep motivated and in their tinkering with regards to the end-assignment. The extensiveness of the OpenCV tutorial was also preferred over the freedom the OFxGui offered.

When asked specifically about the tutorials the students' reactions were mixed. Although all students liked the availability of the tutorials, they clearly stated both forms were useful in their own way. The OpenCV tutorial fitted in neatly with the need for using a library in the end-assignment and the natural flow of the course. OFxGui, partly because it was much more self-study, was simply too easy to put off initially. OFxGui however made its comeback towards the end. As students were looking for a library to include, OFxGui was easily adaptable to the rest of the assignment. Many students, that last minute needed a library, simply adapted OFxGui to fit in. It does however have to be said that while assessing the end assignment most students that used OFxGui like this did not have enough complexity to obtain a grade above a 6.0.

Finally, we confronted the students with the results from the questionnaire. When told about the significant drop in student motivation, the students laughed. "off course it did". According to the students, motivation tends to go down in the middle of a course, due to deadlines and intermittent tests. When conducting the interviews, the night before the oral exams, students said to be a lot more motivated. While working on the end assignments students said to have 'seen the light' and be encouraged by the thrill of utilizing the knowledge acquired in practise.

Overall, the students preferred the more structured provided by the OpenCV example. Utilizing the Flipped classroom method to its full extent simply makes it too easy to put work off. Students do feel they are aided by deadlines and mandatory assignments. Structure provides a platform for students to excel, it helps students to focus and put in the work that is necessary to obtain the required level of competence.

USE OF LIBRARIES IN END-ASSIGNMENTS

Finally, the use of libraries has also been collected together with the grades students obtained during the oral exams. As the use of specific libraries (or similar other ones) was an integral part of the scoring system we will not compare grades over different library groups. Most students implemented at least one library in their end-assignment. Unfortunately, some students still did not use any library, mainly because they perceived this was not in their ability. Luckily this was only about 12% of the students. The other 88% of the class utilized at least 1 library, some even upgraded this to combining up to 3 libraries.

Overall most students implemented a form of computer vision (12 out of 40). Most of these 12 programmes borrowed the use of `OFxOpenCV` as was described in the prototype but some students even managed to get `OFxKinect` running. The end-assignments that utilized Computer vision differed from colour-tracking up to implementing the `CvHaarFinder` and all showed implementations similar to the tutorial that was given.

Another library that was used quite extensively was the `OFxGui`, which was available to students in the form of the second prototype tutorial. It has to be noted that the full Flipped Classroom approach did lead to some undesired effect. Most students that implemented the `OFxGui` did add very little extra complexity, and where mostly just copying the example code from the tutorial into their code in order to tick-off the requirement of using a library. This resulted in quite a low complexity of code within the setting of most assignments that utilized `OFxGui`. As `OFxGui` was also the last tutorial students were given this reflects that most students never fully invested their time into it.

Other libraries were also used to quite an extent; 45% of the students implemented a library that was not even mentioned in the State-of-the-Art research performed in this thesis. Overall this is considered very good as it proves that Creative Technology students implemented their newly obtained knowledge in order to use something completely else. The libraries greatly varied in complexity, but all amounted to overall sufficient grades.

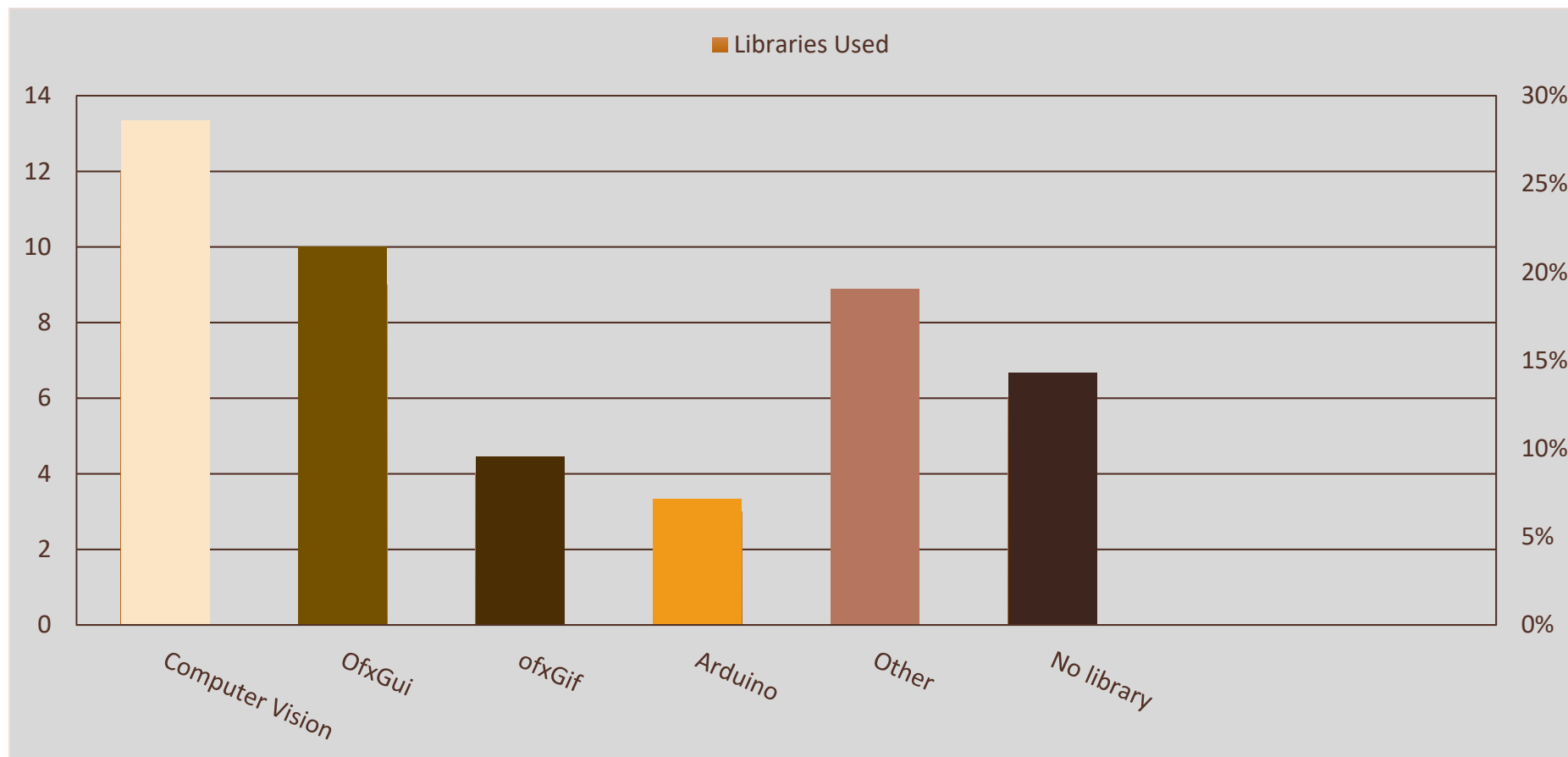


Figure 9; Utilization of the Addons.

STAFF

In order to compare the performance of this year's course with the previous years it has been held the recurring staff (Teachers and more experienced Student assistants) has also been interviewed. They reported a spike in the utilization of both the implemented libraries. Although many students still relied on the use of Arduino for the required physicalization, the increased use of OpenCV did have effect on the total amount of Arduinos seen in the end assignments.

Overall about 30% of students' submissions still did not show enough complexity. These students tried to pass the course with a bare minimum utilization effort, utilizing mainly recycled example code that they were given during the course. Most of these students blamed unclarity in what would be regarded as a library and what would not. Next to them, about 15% of the students showed off excellent work and thus rose significantly above what was minimally required from them.

The previous time the course was given the end assignment was not mandatory. This made it very hard to compare the achieved results from this year with the achievements obtained during the years before. The head teacher from 2 years ago however did notice a trend. In earlier courses quite a lot of students used the Kinect platform in their end assignments. This unfortunately diminished somewhat in the last 2 years the course was given. However, the introduction of the OpenCV tutorial sparked new life into students using a form of Computer Vision (e.g. Kinect, OpenCV). The teaching staff hopes to see this spark create new life for Computer Vision within the Creative Technology program.

All in all, the staff was satisfied with the performance of the tutorials as it did lead to an improvement regarding the use of libraries in general. Although making the use of them mandatory is somewhat of a rough remedy, it did prove effective and the designed tutorials had the intended effect on the student.

With regard to which tutorial was better, no clear answer could be found by the staff. Using both libraries as they were intended had the desired effect. Most students were able to implement a library correctly. This was the intended goal for the staff, and this has been obtained. Overall the staff liked both tutorials and their output, they would use both tutorials next year (with some minor tweaks) and had the impression that the introduction of the tutorials improved the use of libraries.

CHAPTER 9 CONCLUSIONS

PERFORMANCE OF THE DESIGNED COURSES

Both the student and teaching bodies were overall content with the implementation of both library tutorials. Students found the tutorials helpful and allowed them to implement the libraries in a successful manner. The staff was pleased by the improved utilization and creative solutions provided during the end assignments. Both tutorials did however have a unique style and in doing so proved to be suited for their specific target group.

The OpenCV tutorial required more work and was regarded as a harder concept. The students that finished this tutorial showed they mastered a good basic understanding and were likewise rewarded for their efforts in their grades. On the other hand, these students did require more assistance and it was less suited for self-guided use. In the design of the tutorial the required support has been used as a guiding factor. OpenCV has been implemented using the new Guided Read-in concept because it fitted well.

The OFxGui tutorial was mainly used by students that did not attend the tutorial sessions and proved a better option for self-guided learning. The easily available learning content and more intuitive concept helped many students to still incorporate the library in their end assignment in a later stage. Again, due to this reason the original Flipped Classroom methodology has been chosen as the form of implementation.

The complexity of the two libraries has been matched with the structure of the teaching methods. The OpenCV tutorial required more exploration which fitted better to the Guided Read-in method. Likewise, the OFxGui had an already available online tutorial documentation which made it ideal for Flipped classroom. Both methodologies have clear pro's and con's but comparing them with respect to each other is quite impossible. Both methodologies are suited for completely different purposes and both exist within the context of introductory programming education.

Regarding the Flipped classroom principle, conclusions can be drawn about its suitedness for academic purposes. Courses with quite some freedom in working towards the exams are less suited for the Flipped classroom approach. This is because introducing Flipped classroom does require intermitted deadlines to keep students on track. This could be the explanation why it works so well in secondary school, as students there have weekly homework assignments. A very easy way to introduce these intermittent deadlines is by establishing tutorials that are mandatory to sign off.

The Guided Read-in itself is a very convenient method of introducing difficult principles while still using the Flipped classroom approach. It allows students to introduce themselves into the topic before a lecture, while during the lecture more time can be spent on more advanced principles.

Finally, when designing a tutorial regarding libraries the overall style has to be taken into account. Flipped classroom will not be a good approach for all types of libraries. Whenever new principles have to be introduced to the students it is advisable to supply them with a Read-in first and afterward hold a lecture that introduces them to the topic. If the library is quite self-explanatory students will be better off with a complete Flipped Classroom approach. When introducing students to the Flipped classroom approach, the teacher has to keep in mind that deadlines do help students to keep on track and that not all students react toward the Flipped classroom approach in the same manner.

CHAPTER 10 FUTURE WORK

In doing this research one main concern arose. Currently there is quite some research going on improving the educational system. This thesis however encountered a problem that had not been addressed yet. The impact of the detailed structure of a module can be quite severe. During this research the motivation of students went down not due to the course itself but due to the statistics exam in the same week. Later during the course students replied that their motivation had risen again (during the panel interview). This occurred after all other deadlines had past, so students only had to focus on programming. This raises the question to what extent the impact of a course planning influences students' motivation and their performance. Hence course planning (or better yet module planning) should be a major topic, it however is never given its proper place in research.

This research has noticed that the motivation students have is very fluctuating and highly influenced by the impact scheduling has. During the week all students had their statistics exam most students didn't show up, those who did were mainly learning for statistics. It's can be understood that students focus their work on whichever deadline is in front of them, but this also has a major impact on how they perceive other courses.

As a future research topic, I would suggest investigating student motivation in a continues domain throughout an entire module. This may help to understand how students behave through-out a course and what influences their ability to be a 'good' student. This may sound quite trivial, but it could lead to understanding the impact made by decisions taken by the teachers. This impact is very crucial as its results are reflected in the grades obtained by all students. In education, the butterfly effect may be of greater importance then has been considered so far.

FEEDBACK ON THE COURSE PROGRAMMING FOR AI

After the conclusions taken from this research is that there are still quite some things left unsaid. During the research many opportunities for improving the course have been brought to the foreground. These have mostly been left aside in the research due to relevance. These points of feedback are however very thought-provoking and definitely worth to be covered. Hence, they will be mentioned here.

Most students that supplied us with feedback have told us that they appreciated the effort put into improving the course. With regards to this they had quite a lot of respect and were more willing to help. Keeping students involved in the process of improving education helps and should not be forgotten in future research endeavours.

Students are also more engaged with a course when it is neatly structured and mandatory to perform. This last part may sound somewhat childish for a university study but having a guiding structure of scheduled assignments helps student to keep on track. Programming for AI 2018 made the assignments part of the evaluation. This motivated students to perform well on these evaluations to get good grades but paid off in the long run as these students had a better understanding of the course material. It was also mentionable that attendance took a dive after students finished with these tasks.

Furthermore, the availability of teaching assistance was highly valued among the students. Some however did complain they had to wait quite long in order to get the required assistance. A solution for this has been provided by the student panel which suggested using a sign-up system that ensured students neatly queued up without them have to physically stop working.

Finally, in doing the state-of-the-art research regarding libraries, a problem was encountered regarding the OpenFrameworks platform. Frankly, OpenFrameworks is quite out of date and there is very little assistance on the web. One of the things mentioned was simply trying to reform the course in such a way it could be taught in python or another 'newer' programming language.

REFERENCES

- Anderman, E. M., Sinatra, G. M., & Gray, D. L. (2012). The challenges of teaching and learning about science in the twenty-first century: Exploring the abilities and constraints of adolescent learners. *Studies in Science Education*, 48(1), 89-117. doi:10.1080/03057267.2012.655038
- Bishop, J. L., & Verleger, M. A. (2013). *The flipped classroom: A survey of the research*. Paper presented at the ASEE national conference proceedings, Atlanta, GA.
- Brand-Gruwel, S., Wopereis, I., & Walraven, A. (2009). A descriptive model of information problem solving while using internet. *Computers & Education*, 53(4), 1207-1217. doi:10.1016/j.compedu.2009.06.004
- Chan, P. E., Graham-Day, K. J., Ressa, V. A., Peters, M. T., & Konrad, M. (2014). Beyond Involvement: Promoting Student Ownership of Learning in Classrooms. *Intervention in School and Clinic*, 50(2), 105-113. doi:10.1177/1053451214536039
- Conley, D. T., & French, E. M. (2014). Student Ownership of Learning as a Key Component of College Readiness. *American Behavioral Scientist*, 58(8), 1018-1034. doi:10.1177/0002764213515232
- Cunningham, R., Sanjuan Espejo, P., Frederick, C., Sun, L., & Ding, L. (2016). A Second Language Acquisition Approach to Learning Programming Languages.
- de Boer, V., & Winnips, K. (2015). Flipped classroom at the University of Groningen.
- Gommer, E. M., Hermsen, E. M. P., & Zwier, G. (2016). FLIPPED MATH, LESSONS LEARNED FROM A PILOT AT MECHANICAL ENGINEERING. 11.
- Hawi, N. (2010). Causal attributions of success and failure made by undergraduate students in an introductory-level computer programming course. *Computers & Education*, 54(4), 1127-1136. doi:10.1016/j.compedu.2009.10.020
- Jackson, S. (2012). 3 New Teaching Methods Improve the Educational Process. Retrieved from <https://www.gettingsmart.com/2012/09/3-new-teaching-methods-improve-educational-process/>
- Kay, J., Barg, M., Fekete, A., Greening, T., Hollands, O., Kingston, J. H., & Crawford, K. (2000). Problem-based learning for foundation computer science courses. *Computer science education*, 10(2), 109-128. doi:10.1076/0899-3408(200008)10
- Kirschner, P. A., & van Merriënboer, J. J. (2013). Do learners really know best? Urban legends in education. *Educational psychologist*, 48(3), 169-183. doi:10.1080/00461520.2013.804395
- Kölling, M. (2010). The greenfoot programming environment. *ACM Transactions on Computing Education (TOCE)*, 10(4), 14.
- Kuh, G. D. (2003). What we're learning about student engagement from NSSE: Benchmarks for effective educational practices. *Change: The Magazine of Higher Learning*, 35(2), 24-32.
- Mader, A., & Dertien, E. (2016). *Tinkering as method in academic teaching*. Paper presented at the DS 83: Proceedings of the 18th International Conference on Engineering and Product Design Education (E&PDE16), Design Education: Collaboration and Cross-Disciplinarity, Aalborg, Denmark, 8th-9th September 2016.
- Malmberg.

- Melanson, D. (2013). Eric Schmidt: Google now at 1.5 million Android activations per day. Retrieved from <https://www.engadget.com/2013/04/16/eric-schmidt-google-now-at-1-5-million-android-activations-per/?guccounter=1>
- Nunnally, J. (1978). Psychometric methods. In: New York: McGraw-Hill.
- Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., . . . Paterson, J. (2007). A survey of literature on the teaching of introductory programming. Paper presented at the ACM sigcse bulletin.
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer science education*, 13(2), 137-172. doi:10.1076/csed.13.2.137.14200
- Solomon, J. (2005). Programming as a Second Language. *Learning & Leading with Technology*, 32(4), 34-39.
- Teach.com. Teaching Methods. Retrieved from <https://teach.com/what/teachers-know/teaching-methods/>
- Universiteit Twente. (2019). BACHELOR'S PROGRAMME CREATIVE TECHNOLOGY. Retrieved from <https://www.utwente.nl/en/education/bachelor/programmes/creative-technology/>
- Universiteit Twente, T. (2013). Onderzoeken/ ontwerpen/ organiseren: Het Twents Onderwijsmodel. Retrieved from <http://www.utwente.nl/onderwijsvernieuwingen/>
- Vihavainen, A., Paksula, M., & Luukkainen, M. (2011). *Extreme apprenticeship method in teaching programming for beginners*. Paper presented at the Proceedings of the 42nd ACM technical symposium on Computer science education.
- Voogt, J., & Roblin, N. P. (2010). 21st century skills. *Discussienota. Zoetermeer: The Netherlands: Kennisnet*, 23(03), 2000.
- Wells, J., Barry, R. M., & Spence, A. (2012). Using video tutorials as a carrot-and-stick approach to learning. *IEEE Transactions on Education*, 55(4), 453-458. doi:10.1109/TE.2012.2187451

APPENDIX A THROUGH N

A - STAFF INTERVIEWS

Alma Schaafstaal

Wat is een CreaTe'r?

Er is een enorme diversiteit en daardoor is het antwoord wat flauw, maar een CreaTe'r is iemand die afstudeert aan Creative Technology. Oke, dat is te flauw, maar dat komt omdat je er heel veel over kunt zeggen. Uiteindelijk gaat het aan de engineering kant over informatica en elektrotechniek maar het gaat ook over design, het gaat ook over entrepreneurship, er zit een basis van onderzoek in en samen leidt dat tot een soort student die afhankelijk van zijn eigen liefdes heel ver is in bepaalde delen en minder ver in andere delen. En dat leidt tot een hele diverse student met een liefde voor het maken van dingen. Ze hebben absoluut geen angst met het aanpakken van technologie en dat kunnen ze ook, ze zullen ook gewoon een soldeerbout oppakken en iets gaan doen. Ze willen ook dingen maken die er toe doen, het feit dat een project verder gaat in demo markten en dingen als gogbot. Ze zouden wel eens wat meer zelfvertrouwen mogen hebben, al hebben alle opleidingen wel dat studenten soms wat zelfvertrouwen missen. Echter wordt dat effect versterkt door het feit dat de opleiding zo breed is.

Waar hebben we CreaTe'r voor nodig?

Nou ze zijn wel echt de engineers van de toekomst omdat de toekomst wel vraagt om mensen die multidisciplinair problemen op kunnen lossen. Het vraagt om mensen die kunnen meekijken in meerdere disciplines en daarbinnen kunnen schakelen en kunnen begrijpen zonder daar dezelfde hoeveelheid diepte in te begrijpen. Mensen met een Create achtergrond zijn tevens ook in staat onorthodoxe oplossingen te verzinnen. Want je kunt wel ergens gewoon een bestaande oplossing voor verzinnen maar dat mist vervolgens wel het 'wow' gevoel. En daarvoor heb je mensen nodig die over problemen heen kunnen kijken om tegelijkertijd met hun combinatie van kennis in staat zijn om die nieuwe onorthodoxe oplossingen te bedenken.

Wat is belangrijk voor Create'rs om gave dingen te bouwen?

Op dit moment denk ik dat data science heel hip en hot is en dat een Create'r hier vooral datavisualisatie en dataphysicalisatie aan toe kan voegen. Maar ook CyberSecurity gezien door een Create'r hele interessante dingen kan opleveren.

Wij als CreaTe'rs moeten eigenlijk beyond the apps gaan werken en daarvoor is het eigenlijk jammer dat we niet in staat zijn fatsoenlijk app's te maken. Een app is op zich nog wel interessant als het iets anders bedient maar gewoon een app is niet zo interessant. Is een beetje flauw, geld namelijk echt niet voor alles, maar dat een app op zich de oplossing is geloof ik niet meer in, dat is geweest. Maar de data-analyse icm wearables en wat kunnen we nou allemaal zien/meten vind ik een stuk interessanter.

Spraakmakend gebruik van de Hot Topics?

De meetoo datavisualisatie omdat hij maatschappelijk veel impact heeft en er niet wordt weggelopen van het onderwerp. Er zit heel veel kracht in die visualisatie. Wat gedaan is met het Ovho was ontzettend interessant omdat het heel mooi kan zijn maar vooral ook alle data-analyse die mogelijk zou kunnen zijn met wat we tot nu toe gedaan hebben. De combinatie van hardware en software blijft ook erg interessant omdat ze best veel overtuigingskracht en fascinatie oplevert doordat er echt iets fysieks voor je staat te gebeuren. En daarbij gaat het vooral over de combinatie van dingen, daarvan weet ik niet zo goed wat je nou nodig hebt. Datascience is natuurlijk ook gewoon een heel groot onderwerp waar we veel mee kunnen gaan doen. Zo komen er grote slagen met combinatie van sport of the wearables maar dan moeten we er wel echt opduiken om daar dingen mee te gaan/kunnen doen.

Andere dingen die ik erover kwijt wil

Wat voor mij nog meer ontzettend belangrijk is is dat er rekening wordt gehouden met onze instroom. Er zijn een aantal mensen dat binnen komt dat informatica hun struikelblok gaat worden en er moet ook bezig gegaan worden met het feit dat de mindset blijft dat we programmeren kunnen en dat het nog leuk is ook. En niet iedere docent in dit gebouw die gewend is te werken met harde Bèta studenten kan dat handelen. Hij begrijpt niet perse dat studenten daar aarzelingen bij kunnen hebben. Dingen zoals de sportsday maken het playfull en zorgt er voor dat mensen de ruimte hebben het echt toe te passen in iets werkelijks. Zodat er tinkering-wise mee verder gegaan kan worden. Als wij als Create meer mensen daar de technische vakken willen trekken moeten we actief bezig gaan met het overkomen van die mindset dat het eng of moeilijk zou zijn. Het is immers niet voor niks dat ansgar en Angelika die prijs hebben gekregen voor tinkering. Die kracht sleept stiekem gewoon heel veel mensen door de wiskunde en het programmeren heen die er anders nooit doorheen waren gekomen. Maar je moet ook je best doen om de betere studenten aan de bovenkant van het spectrum ook geboeid blijven met en onderwijs en niet weglopen met het idee dat het te simpel is. Je moet bij zo'n diverse groep gewoon heel erg nadenken over de didactiek die je toepast.

Richard Bults

Wat is een Create'r?

Ik denk dat dat een van de lastigste vragen is die je kunt stellen. Op zijn minst is een creator iemand die inzetbaar is in multidisciplinaire projecten met een hele brede skillset. Hij is de ontwerper die een brede kijk heeft op alles wat er in de maatschappij gebeurt waar technologie (vooral met een ICT component) een rol in speelt.

Waarom hebben we Create'rs nodig?

Omdat er genoeg specialistische ingenieurs zijn die op een 'enge' manier naar de mogelijke oplossing voor een probleem waar een Create'r juist op een hele brede manier naar een probleem kijkt. En daarmee neemt de kans op een betere oplossing te

Wat wordt verwacht van een Create'r?

Ik zie een creator zitten tussen de probleemeigenaar (klant) en de specialist die het uiteindelijke product kan maken. Als je een informaticus voor de probleem eigenaar zet, zal hij vragen geef mij maar de eisen voor het systeem en ik zal het maken. Een create'r zal vragen wat een klant er nou precies mee wil en vervolgens samen met de klant bedenken wat dan de eisen zijn voor de specialisten. Hij zit als het ware in een soort spilfunctie. De skill-set van de create'r gaat nog verder omdat er een prototyping plaats kan vinden zodat er niet alleen requirements op papier komen te staan maar ook een ontwerprichting kan worden aangedragen.

Wat is belangrijk voor Create'rs om gave dingen te bouwen.

Het maakt niet zoveel uit welke programmeertaal een create'r krijgt, hij moet vooral in functionele structuren leren denken. Hij moet in staat zijn het probleem op te delen in kleinere problemen, enzovoorts. Hij moet in staat zijn een goede decompositie te maken. Vervolgens is het de vraag hoe ieder probleem in logische functies gaat worden opgelost met software dan wel hardware. En dat is voor mij het belangrijkste, als jij namelijk in staat bent om een functionele structuur te maken voor je systeem maakt het namelijk niet (zoveel) uit in welke programmeertaal je dat moet gaan implementeren. Het is veel meer de manier van denken dan de daadwerkelijke implementatie.

Welke Tools/Libairies zijn nuttig voor Creative technologies?

Wat je meteen ziet is dat er nog al veel create'rs zijn die niet zo happig zijn op een afstudeeropdracht (GP) met een programmeer component. Waar docenten echter wel om vragen is het gebruik van mobiele applicaties. Voor IOS is dat

een lastig verhaal dus is de Android omgeving veel interessanter voor het onderwijs. Als we dat met zijn allen willen heeft dat wel een positief effect op de GP's die create'rs kunnen doen. Dus dat is zeker een richting.

Verder denk ik dat vaardigheid in python erg belangrijk is omdat dit bijna platform onafhankelijk is. Tot op zekere hoogte kun je een python programma op je laptop ontwikkelen en die bijna (tot op zekere hoogte) een op een overzetten op een microcontroller (raspi). Dat geeft heel veel flexibiliteit. Student moet ook in staat zijn om te ontwikkelen met arduino's omdat die component zwaar in het create curriculum zit. Ik zou het jammer vinden als we create'rs alleen leren ontwikkelen op het arduino platform. Ik zou zelf liever inzetten op Python omdat het veel volwassener is met een veel groter community platform. We scoren wel eens in de VR wereld maar dat is Unity Programming. Alles wat met de interfacing van de wereld te maken heeft gehad zoals smart city measurements (serious) Game-development komt ook steeds meer op binnen de opleiding. AR komt ook op, maar daar hebben we helaas nog weinig support voor. Omdat het wel veel kan betekenen voor de maakwereld.

Erik Faber

Wat is een Create'r?

Ik kan je denk ik niet een kort en bondig antwoord geven over wat een Create'r nu eigenlijk is, dus ik zal een beschrijving moeten geven. Hij is ten eerste heel multidisciplinair, eerder een generalist dan een specialist omdat je met Creative natuurlijk nooit echt de diepte in gaat. We zetten onszelf neer als een Ontwerpersopleiding, omdat een heel groot deel van wat je leert aan skills binnen je opleiding zich focused op de ontwerpmethodologie en User Centered design. Dat vind ik echt een asset ten opzichte van andere technische opleidingen omdat je in staat bent iets te ontwikkelen in samenspraak met de gebruiker zodat je niet in je eigen cocon iets ontwerpt en dan maar hoopt dat de gebruiker het ook leuk vindt. Dat is een aspect, je wordt een ontwerper. Verder is de wereld waarin we dat ontwerpen leren heel duidelijk de combinatie van elektrotechniek en informatica en we hebben het dus altijd over intelligent systems.

Waarom hebben we Create'rs nodig?

Omdat de hoeveelheid technische skills gewoon steeds verder toeneemt, er kan steeds meer en we zijn dus niet meer in staat iemand alles te leren. Het curriculum van bijvoorbeeld elektrotechniek moet echt al in zijn basisvakken snijden omdat het anders niet in een basisvak past. Dus zometeen heeft er niemand meer een overzicht van wat is er nou precies nodig, waardoor je steeds verder van de klant weg drift omdat je jezelf verliest in de mogelijkheden van de technology. En daar zijn Create'rs voor nodig die vanuit hun ontwerpers kant bruggen kan slaan tussen de gewone mens die moet kunnen werken met de technologie en aan de andere kant technische ontwerper. Daartussen zit een heel groot gebied waar maar weinig studies echt op inspelen. En Creative Technology blinkt uit in het meegeven van tools uit dat gebied daar tussenin. Zodat je zowel de taal leert om de gebruiker te kunnen begrijpen en zijn requirements om te zetten in iets wat wordt begrepen door de echte technische achterban. En Create'rs worden ontzettend belangrijk omdat we steeds meer de vraagstukken krijgen rondom waar we technologie voor gaan inzetten.

Wat wordt eigenlijk verwacht van een Create'r?

We hebben het hier over bachelors en nog niet masters maar studenten moeten laten zien dat ze een open vraagstuk kunnen oplossen doordat ze zelf een plan kunnen ontwikkelen vanaf de wensen van de klant tot aan het prototype dat aantoont dat iets kan werken in de praktijk. Hij moet kunnen laten zien dat hij zijn ontwerp skills heeft en die kan toepassen maar dat hij ook de technische know-how heeft om zoiets te bouwen

Wat is belangrijk voor Create'rs om gave dingen te bouwen En daarnaast de hot topics?

Het mooie aan Create is dat we de studenten de vrijheid geven te kiezen waar ze mee willen werken. Dit hebben we al met New Media / Smart Tech maar daarnaast ook in waar ze mee willen werken. Het is dus niet mogelijk om hier een omvattend antwoord op te geven omdat naast Module 5 ze ook andere tools vergaren binnen hun minor. Programmeren echter is een verplicht onderdeel dat iedereen moet kunnen beheersen. Er blijven echter studenten die bij hun afstuderen

aangeven liever een opdracht te hebben met zo min mogelijk programmeren. En we willen niet dat studenten niet hun technische know how laten zien. The smart rainwater buffer, want daar zitten meerdere sensoren in het Data Processing en het visualiseren van die data. Want informatica geeft bijzonder weinig aandacht aan datavisualisatie maar voor een CreaT'r is dat echt wel een asset. Maar ook het classificeren van data, voor zover dat binnen programmeren valt, kun je bijvoorbeeld beweging interpreteren of andere vormen van signaalanalyse. Soms komen we er toch achter dat, ondanks dat CreaT'rs geen app-bouwers zijn, ze toch iets met een app moet bouwen, daar hebben ze nog niet echt kaas van gegeten maar de meeste pakken dat toch wel goed op.

Kasia Zalewska

What is a Create'r?

Yeah, that's a very difficult question, after the study; I think a creat'r is somebody who has ideas, is creative, and has knowledge about different domains, technology and design. Maybe not so much in depth, as he has to know a lot about different things. But he has the knowledge to test if a product would work and about how it's perceived by the users. For me, a creat'r is someone that takes people into consideration, he doesn't create technology for the sake of technology but more for the benefits of people for either enjoyment but also for issues like for instance healthcare, that doesn't matter. He also has to be able to communicate; he has to be able to talk with the consumer but also tell the engineers how to create the solution.

Why we need CreaT'rs?

First of all for innovation; without creative people that understand technology and the needs of consumers there wouldn't be any progress. To me CreaT'rs are also very specific people as they are different compared to technology students or other students. They do geeky stuff but they are not geeks per se. They all have a lot of interest beyond their study and I think that is what makes them special as they can reconfigure all that stuff.

What is expected of a CreaT'r?

Let's take games as an example; A lot of students are interested in games. If you want to create a game you need to know how much work it is and how to design the process to create that game. You will need people that create the storyline and the characters and 3D modelers and programmers that code it all together. And because there are a lot of products that need programming. Because of that I think in a couple of years every kid needs to learn programming. Because it's needed in the current society, I'm now wearing a one purpose watch but most colleges are wearing smart-watches. And there is so much you could do with a smartwatch, for which you need CreaT'rs

What are hot topics for a CreaT'r

Smart cities and what it is really and how municipalities can get invested in it. CreaT'rs can get into all the steps surrounding this project like the data visualizations, users interfaces, what to use the technology for and how to get people involved and they can do this in a creative way. Like a computer scientist could program this all better (more structured, more efficient) but it will probably be boring (brings nothing new to the table). I also believe CreaT'rs can help in establishing that all kids learn programming in an easy and fun way, and how you would do that.

Basically everything with smart-, because they are also trained in the ethics surrounding it all. And what impact technology will have on people. They can be a sort of buffer for cyber security as they know what the impact might bring before (normal) people [users] realise it. They really bring the human side and technology together.

Robby van Delden

Wat is een Create'r?

Is iemand die nieuwe technologieën op een creatieve manier samengevoegd, vaak in een nieuw toepassingsdomein waardoor er mogelijkheden worden gecreëerd die daar vroeger niet waren.

Waarom hebben we Create'rs nodig?

Omdat het in deze wereld vaak gaat over het kunnen zien van die nieuwe mogelijkheden en dat vaak niet gedaan word. En met een opleiding als CreaTe die heel breed is en veel variatie tussen de mensen dit wel word gestimuleerd

Wat wordt eigenlijk verwacht van een Create'r?

Dat ze mogelijkheden zien; Dat ze in staat zijn een functionerend prototype in elkaar te 'hacken'. Dat ze daarmee verder gaan dan wat het eerste idee was; Dat ze kunnen luisteren naar de eindgebruiker/opdrachtgever en daarmee tot aanpassingen en requirements kunnen komen. Maar ook dat dat ze op nieuwe ideeën brengt voor verbeteringen. Waar ze uiteindelijk terecht komen varieert heel erg tussen de interesse van de individuele student.

Wat is belangrijk voor Create'rs om gave dingen te bouwen.

Sirius gaming maar dat ligt wat lastig vanwege tweedeling mod 5, Verder zie ik niet echt gaten in het curriculum

Wendy Oude Nijeweme

wat is een Create'r?

Een CreaT'r weet een probleem op creatieve wijze een goede oplossing op te verzinnen, door heel erg out of the box te kunnen denken. Ze zoeken niet alleen een oplossing maar spelen daarbij ook in op de behoefte van de mensen.

Waarom hebben we Create'rs nodig?

Omdat ze oplossing binnen een casus bedenken waar ik niet in eerste instantie aan zou denken zowel binnen de wat meer voor de hand liggende oplossing als hele andere oplossingen waar nog niemand aan gedacht heeft

Wat wordt eigenlijk verwacht van een Create'r?

Sowieso een goed overzicht van de nieuwste technieken en dat altijd blijven updaten maar daar moeten ze ook mee kunnen werken en de goede mensen daarin blijven aansturen. Hierin is de communicatie tussen de gebruikers en de techneuten ontzettend belangrijk. Om bruggen te leggen tussen hoe bijv. Artsen denken en hoe de programmeurs dat doen. Zeker het zijn van die brug is een erg belangrijke functie van CreaT'rs in de wereld

Wat is belangrijk voor Create'rs om gave dingen te bouwen, hot topics?

Ik denk dat het heel erg belangrijk is om de interesse van de CreaT'r mee te nemen in de behoefte. Iedereen gaat zich voor een project nog extra verdiepen en dat is veel bepalender voor wat een Hot Topic is. Verder is IoT en ML erg belangrijk voor creative technology.

Ansgar Fehnker

Wat is een Create'r?

De essentie van een CreaT'r is dat hij weet hoe hij met technologie om moet gaan en daar iets moois/ nieuws van maakt. Maar vooral de competentie om met technologie om te gaan en daar niet bang voor te zijn

Waarom hebben we Create'rs nodig?

Er zijn al genoeg mensen die echt experts zijn, die ver de diepte in gaan, maar vaak ontstaan nieuwe dingen door hercombinaties van dingen die al bestaan of net nieuw zijn met iets bestaands en daarvoor moet je creatief kunnen recombineren en daar denk je niet aan als je al in de diepte zit. Je moet het lef hebben om het gewoon uit te proberen. A Jack of all trades and master of none. Je moet vooral goed met iedereen kunnen meepraten.

Wat wordt eigenlijk verwacht van een Create'r?

Dat ze uitdragen dat je positieve dingen kunt doen met technologie, dat ze kritisch zijn maar de maatschappelijke drempel wel kunnen verlagen. En ze moeten Creatief/ kunstzinnig kunnen zijn met design en daar wel een gevoel bij hebben. Wat is belangrijk voor Create'rs om gave dingen te bouwen. Je moet weten hoe je een manual kunt lezen en kunt googlen maar je er ook soms niet aan te houden. Niet bang zijn om dingen open te schroeven. Je moet vooral niet bang zijn om dingen te doen/ondernemen en het maar gewoon uit te gaan proberen.

Welke Tools/Librairies zijn nuttig voor Creative technologies?

- De overstap kunnen maken naar een IoT infosysteem
- Beeldverwerking
- Signaalverwerking
- Language processing
- Mobile devices

Wat wordt er aanbevolen

- Data meer toegankelijk maken
- Mensen bekender maken met dingen
- Real life games (rpg met techniek)
- Body Area Networks

Angelika Mader

Wat is een Create'r?

Een creator is iemand die nieuwe problemen kan identificeren, hij kan ze framen en daar oplossingen voor bedenken. Hij kan dus ook met slecht geformuleerde problemen omgaan en probeert zo het dagelijkse leven van mensen te verbeteren.

Waarom hebben we Create'rs nodig?

We hebben ze nodig voor een bredere mindset. Overal duiken discussies op over dat monodisciplinaire studenten nog steeds wel nodig hebben maar dat de vraag steeds meer opkomt voor studenten die een breder gereedschap tot hun beschikking hebben. In bedrijven willen ze ook mensen die wat breder kunnen kijken, met andere goed kunnen communiceren en slecht gedefinieerde problemen kunnen oplossen. Dat hoort bij de 21st century communications en daar zijn onze studenten goed in.

Wat wordt eigenlijk verwacht van een Create'r?

Wat is belangrijk voor Create'rs om gave dingen te bouwen.

Hij heeft echt een goede technologische toolkit nodig hebt. We hebben er tussenzitten die zich echt focussen op alleen concepten maar ik denk dat dat tamelijk zwak is. Een echt goede creaT'r is openminded, heeft een visie en bezit ook andere doelen in het even dan alleen design, hij wil ook aan de slag met technology

Welke Tools/Librairies zijn nuttig voor Creative technologies?

Sustainability,

Mensen helpen omgaan met Data,

Ik zou vooral iets kiezen wat speels en waardoor de vonk over kan slaan, we kunnen toch niet alles behandelen dus studenten moeten dit vooral zelf durven oppakken als het nodig wordt.

B – EVALUATION RUBRIC

Evaluation form Programming - module 6 .1.2018						
student name:	examiner:					
student name:	assistant					
short description:						
installation:						
working installation:.....	<table border="1" style="border-collapse: collapse; width: 100px; height: 25px;"> <tr><td></td><td></td><td></td><td></td><td></td></tr> </table>					
smooth interaction:.....	<table border="1" style="border-collapse: collapse; width: 100px; height: 25px;"> <tr><td></td><td></td><td></td><td></td><td></td></tr> </table>					
complexity of application:.....	<table border="1" style="border-collapse: collapse; width: 100px; height: 25px;"> <tr><td></td><td></td><td></td><td></td><td></td></tr> </table>					
AI algorithm adequately included:.....	<table border="1" style="border-collapse: collapse; width: 100px; height: 25px;"> <tr><td></td><td></td><td></td><td></td><td></td></tr> </table>					
library used:.....	<table border="1" style="border-collapse: collapse; width: 100px; height: 25px;"> <tr><td></td><td></td><td></td><td></td><td></td></tr> </table>					
creativity of the application:.....	<table border="1" style="border-collapse: collapse; width: 100px; height: 25px;"> <tr><td></td><td></td><td></td><td></td><td></td></tr> </table>					
attention for details:.....	<table border="1" style="border-collapse: collapse; width: 100px; height: 25px;"> <tr><td></td><td></td><td></td><td></td><td></td></tr> </table>					
structure:						
classes adequate:.....	<table border="1" style="border-collapse: collapse; width: 100px; height: 25px;"> <tr><td></td><td></td><td></td><td></td><td></td></tr> </table>					
functions/methods adequate:.....	<table border="1" style="border-collapse: collapse; width: 100px; height: 25px;"> <tr><td></td><td></td><td></td><td></td><td></td></tr> </table>					
global/local variables adequate:	<table border="1" style="border-collapse: collapse; width: 100px; height: 25px;"> <tr><td></td><td></td><td></td><td></td><td></td></tr> </table>					
documentation/ meaningful code:						
header:.....	<table border="1" style="border-collapse: collapse; width: 100px; height: 25px;"> <tr><td></td><td></td><td></td><td></td><td></td></tr> </table>					
comments (not stating the obvious):.....	<table border="1" style="border-collapse: collapse; width: 100px; height: 25px;"> <tr><td></td><td></td><td></td><td></td><td></td></tr> </table>					
can explain/answer questions:.....						
<table border="1" style="border-collapse: collapse; width: 100px; height: 25px;"> <tr><td></td><td></td><td></td><td></td><td></td></tr> </table>						

C – OFXGUI TUTORIAL / FLIPPED CLASSROOM

This tutorial is based on the online lectures regarding OFxGui, by Lewin Lepton and can be found on YouTube.

https://www.youtube.com/watch?v=X_is1x8iVtw	Introduction
https://www.youtube.com/watch?v=YSma8FBNRfo	Colours and shapes
https://www.youtube.com/watch?v=a0s93Ey47dQ	Sound
https://www.youtube.com/watch?v=0_xT3ODh7s0	Video
https://www.youtube.com/watch?v=h8i7LYt89tg	ofParameter
https://www.youtube.com/watch?v=J_FIWS5C2wc	Talking to classes

TUTORIAL EXCERSIES

1. Display a polygon on screen and allow its characteristics to be alerted live using OFxGui.
Characteristics;
 - a. colour
 - b. Size
 - c. Position
 - d. Amount of sides
2. Edit you final OpenCV tracker to work with variable amount of;
 - a. Colours
 - b. Threshold
 - c. amount of blobs
3. Generate a playing-field (maze, board, landscape, whichever suits your end-assignment best) which can be adapted life with variables.
Some inspiration; <https://www.youtube.com/watch?v=3thB8uQ-H6I>

D – OPENCV TUTORIAL / GUIDED READ-IN

The example explained below can be found (and run) in your OpenFrameworks root directory; `Of_v*.***.*/examples/(addons/)computer_vision/opencvExample/opencvExample.sln (.xcodeproj)`

I strongly advice you now to open it and give it a test run!

First look at the .h file

```
#include "ofxCv.h"

#define _USE_LIVE_VIDEO           // uncomment this to use a live camera
                                // otherwise, we'll use a movie file
```

So the first two steps in the header file include the specific library and do or do not call the webcam

Further on we find;

```
#ifdef _USE_LIVE_VIDEO
    ofVideoGrabber      vidGrabber;
#else
    ofVideoPlayer       vidPlayer;
#endif
```

Switching the modes between video input and live video. Keep this #define structure in mid because this type of 'if' is used later on as well.

And there are also a couple of objects/variables systematically used in the program;

```
ofxCvColorImage      colorImg;
```

The image that stores the frame(left top corner)

```
ofxCvGrayscaleImage  grayImage;
```

This image stores the frame in grayscale (right top corner)

```
ofxCvGrayscaleImage  grayBg;
```

This image stores the saved background frame (left middle frame)

```
ofxCvGrayscaleImage  grayDiff;
```

This image stores the difference between the Background and the new frame (right middle frame)

```
ofxCvContourFinder  contourFinder;
```

This object stores the found contours after analysis (bottom right)

```
int                  threshold;
```

Threshold value for the analysis

```
bool                 bLearnBakground;
```

Trigger for relearning the background

Next is the .cpp File;

```
void ofApp::setup(){

    #ifdef _USE_LIVE_VIDEO
        vidGrabber.setVerbose(true);
        vidGrabber.setup(320,240)
    #else
        vidPlayer.load("fingers.mov");
        vidPlayer.play();
        vidPlayer.setLoopState(OF_LOOP_NORMAL);
    #endif
}
```

In the Setup we see that the 'if' mentioned earlier is used here to switch between the initialization of either the camera (with a pixel format of 320 x 240) or the video (which loops and uses "fingers.mov").

```
colorImg.allocate(320,240);
grayImage.allocate(320,240);
grayBg.allocate(320,240);
grayDiff.allocate(320,240);
Next to that all our images are set to the same pixelformat
bLearnBakground = true;
make sure initial background is set
threshold = 80;
and set the initial threshold to 80
}
```

In the update we have the usual calls like repainting the background but also more specific commands.

```
void ofApp::update(){
    ofBackground(100,100,100);

    bool bNewFrame = false;

    #ifdef _USE_LIVE_VIDEO
        vidGrabber.update();
        bNewFrame = vidGrabber.isFrameNew();
    #else
        vidPlayer.update();
        bNewFrame = vidPlayer.isFrameNew();
    #endif
}
```

The 'ifdef' is used here again to update the frame and only call the rest of the update-function if the frame is actually a new frame. This is to make sure the computer only needs to redo its calculations "whenever the frame is actually changed. This is done by setting bNewFrame to 'true'.

```

if (bNewFrame){
    #ifdef _USE_LIVE_VIDEO
        colorImg.setFromPixels(vidGrabber.getPixels());
    #else
        colorImg.setFromPixels(vidPlayer.getPixels());
    #endif

```

If the frame is new it will be loaded into a pixel array (an image) in order to process the frame.

```

grayImage = colorImg;

```

This automatically reframes the colorImage into a grayImage (as these have been defined as such)

```

    if (bLearnBakground == true){
        grayBg = grayImage;           // the = sign copys the pixels from
grayImage into grayBg (operator overloading)
        bLearnBakground = false;
    }

```

Whenever the background has just been reset (by hitting the spacebar bLearnBakground is set to true) this is where that frame is stored as the background.

```

// take the abs value of the difference between background and incoming
grayDiff.absDiff(grayBg, grayImage);

```

This allows you to see what the difference between 2 pictures as those parts is light up white while the backgrounds stays white. The larger the difference is the brighter will be the display.

And although that looks very easy it's quite hard to calculate with these new pixelmaps. And thus, we filter them by overlaying a threshold filter. This filter takes white-values of a certain level or higher and rounds them up to 255 while everything below it is rounded down to 0;

```

and then threshold:
    grayDiff.threshold(threshold);

// find contours which are between the size of 20 pixels and 1/3 the w*h pixels.
// also, find holes is set to true so we will get interior contours as well....
    contourFinder.findContours(grayDiff, 20, (340*240)/3, 10, true);
// find holes
    }
}

```

Finally, we analyse the difference image for 'blobs'. 'blobs' is probably the best word in academic programming and refers to areas of white spots within this black image (that resulted from our grayDiff image). This function however requires quite some function inputs.

First of all is the image you would like to analyse (grayDiff in our case),

```

contourFinder.findContours(grayDiff, 20, (340*240)/3, 10, true);,

```

Secondly the minimum size of any 'blob' and maximum size of a 'blob' to control what elements it should or should not focus on.

	Min	Max
<code>contourFinder.findContours(grayDiff, 20, (340*240)/3, 10, true);</code>		

The 4th attribute sets the maximum amount of 'blobs' in the image. All blobs are detected but only the largest Nth 'blobs' are displayed.

<code>contourFinder.findContours(grayDiff, 20, (340*240)/3, 10, true);</code>

And finally, the 'true' allows to program to find holes. The holes attribute checks for 'blobs' within 'blobs'; these are negative patches completely surrounded by a positive patch (where normal 'blobs' are positive patches within the negative backgrounds).

<code>contourFinder.findContours(grayDiff, 20, (340*240)/3, 10, true);</code>

The Draw and Keypressed function should be quite straightforward but do read them and ask your questions at the beginning of the lecture. Furthermore, I now encourage you to play a little with the example code and **come up with 3 things you could use this for?**

- **Tracking one's movement, with accompanying speed**
 - o The person will walk within the frame, and as its direction and speed is presumably constant, the person will walk through the screen in a given time.
 - o The frames will light up, giving you a "framejump" over time, and as you know distance you can calculate its speed
 - Might be helpful for sporters
- **Medical use**
 - o Use the software to check someones reaction speed
 - o the video will be of constant lengths, therefor it has a good reference for seeing how long people do to see change
- **Checking inconsistencies within a video**
 - o Could be used to see if a video is edited, as this will change the setup of the video.
 - o the changed part might happen at a faster rate, so this can be seen as the blobs increase and decrease per edited part.

OpenCV Tutorials

1. **resize();** Let us start simple, you have already encountered in the example how frames are drawn and how the frame-sizes are allocated. Now change this standard format to something you think is more fitting. The last frame should be more fitting for your screen than the 320x240 (I presume you are all working with at least 1280*720) so the user can better observe the end result. Keep in mind that most of the subframes you have can help you with your analysis and debugging and that you still want them to be in your window.

When doing this, make sure you do not get any allocation warnings and the example-code is still able to run.

2. **Bluring,** luckily for us we do not have to design our own blur filters. OpenCV has quite some already build into them which we can use. These are called by functions like

<code>... .blur(i);</code>

For the NewMedia students this should ring a bell, as it represents the following filtering matrix;

$$\frac{1}{9}$$

1	1	1
1	1	1
1	1	1

Where *i* indicates the dimension of the array;
so this one exists for `.blur(3)`;

Test the result of multiple different blurs and blur-values. What would you be able to use this for? Are tracing images getting fuzzier or more clear?

- **.blur(34) does a lot of blurring → tracing gets more fuzzy**
- 3. **ofPixel and Colours**, now we are going to really dive into the image files and try to edit them. For this we first want to convert our `ofCV....Image` into only an array of pixels. In OpenCV for OF this is done by creating a `ofPixel` variable and loading the image in from pixels.
 - a. *Create an variable of the `ofPixels` type and store the `ColorImage` into it via `getPixel()`.*

Now you should be able to retrieve colours at specific spots utilizing

```
... .getColor(x,y);
```

and altering specific spots by utilizing

```
... .setColor(x,y,color);
```

- b. *Alter the image by making your own filter () by changing the colour-values of all pixels individually. For this you could first use `.convertRgbToHsv()`; to covert an `ofCVColorImage` form the RGB scale to the HSV scale which helps comparing colours (<http://colorizer.org/>)*
- 4. **Finally, we are going to build something**; Now you know all the steps to build something and you are going to build a colour-tracker. This means that your filter returns the value of black for all pixels that aren't that specific colour and white (or the colour itself) for all pixels that are that specific colour. Now using the same things have seen in the example code find the contours of your generated image. Finally, use the `contourFinder.blobs[]` objects to draw red dots in the middle of each blob/traced contour.

E – SIMS MOTIVATIONAL SCALE

The Situational Motivation Scale (SIMS)

Directions: Read each item carefully. Using the scale below, please circle the number that best describes the reason why you are currently engaged in this activity. Answer each item according to the following scale: 1: *corresponds not at all*; 2: *corresponds a very little*; 3: *corresponds a little*; 4: *corresponds moderately*; 5: *corresponds enough*; 6: *corresponds a lot*; 7: *corresponds exactly*.

Why are you currently engaged in this activity?

- | | |
|--|---------------|
| 1. Because I think that this activity is interesting | 1 2 3 4 5 6 7 |
| 2. Because I am doing it for my own good | 1 2 3 4 5 6 7 |
| 3. Because I am supposed to do it | 1 2 3 4 5 6 7 |
| 4. There may be good reasons to do this activity, but personally I don't see any | 1 2 3 4 5 6 7 |
| 5. Because I think that this activity is pleasant | 1 2 3 4 5 6 7 |
| 6. Because I think that this activity is good for me | 1 2 3 4 5 6 7 |
| 7. Because it is something that I have to do | 1 2 3 4 5 6 7 |
| 8. I do this activity but I am not sure if it is worth it | 1 2 3 4 5 6 7 |
| 9. Because this activity is fun | 1 2 3 4 5 6 7 |
| 10. By personal decision | 1 2 3 4 5 6 7 |
| 11. Because I don't have any choice | 1 2 3 4 5 6 7 |
| 12. I don't know; I don't see what this activity brings me | 1 2 3 4 5 6 7 |
| 13. Because I feel good when doing this activity | 1 2 3 4 5 6 7 |
| 14. Because I believe that this activity is important for me | 1 2 3 4 5 6 7 |
| 15. Because I feel that I have to do it | 1 2 3 4 5 6 7 |
| 16. I do this activity, but I am not sure it is a good thing to pursue it | 1 2 3 4 5 6 7 |

Codification key:

Intrinsic motivation (IM): 1, 5, 9, 13

Identified regulation (IR): 2, 6, 10, 14

External regulation (ER): 3, 7, 11, 15

Amotivation (A): 4, 8, 12, 16

Self-determination Index (SDI) = 2 x IM + 1 x IR – 1 x ER – 2 x A

F – QUESTIONNAIRE 1

1. Because I am interested in programming *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

2. But I am not sure if it is worth it *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

3. Because I feel good when doing a programming course *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

4. Because I decided I want to learn about OpenFrameworks *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

5. Because I have to pass this course in order to pass the module *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

6. Because I think programming is good for me *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

7. Because I have to follow the course *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

8. Because I think programming is pleasant *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

9. And there may be good reasons to learn OpenFrameworks, but personally I do not see any *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

10. Because I believe being able to program in OpenFrameworks is a valuable skill in my skill set *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

11. Because I am supposed to do it *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

12. Because I feel that I have to do it *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

13. Because it is fun to program *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

14. But I am not sure OpenFrameworks is a good thing for me to pursue *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

15. Because I believe it is important for me *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

16. But I do not see why the course is given in OpenFrameworks *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

I'm learning how to Program Search algorithms...

17. Because I am interested in Search algorithms *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

18. But I am not sure if it is worth it *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

19. Because I feel good when working with Search algorithms *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

20. Because I decided I want to learn about Search algorithms *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

21. Because I have to pass this course in order to pass the module *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

22. Because I think being able to program Search algorithms is good for me *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

23. Because I have to follow the course *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

24. Because I think programming Search algorithms is pleasant *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

25. There may be good reasons to learn how to work with Search algorithms, but personally I do not see any *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

26. Because I believe being able to program with Search algorithms is a valuable skill in my skill set *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

27. Because I am supposed to do it *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

28. **Because I feel that I have to do it ***

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

29. **Because it is fun to program with Search algorithms ***

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

30. **But I am not sure if being able to program Search algorithms is a good thing for me to pursue ***

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

31. **Because I believe it is important for me ***

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

32. **But I do not see why the course teaches Search algorithms ***

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

Thank you very much!

G – QUESTIONNAIRE 2

1. Because I am interested in programming *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

2. But I am not sure if it is worth it *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

3. Because I feel good when doing a programming course *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

4. Because I decided I want to learn about OpenFrameworks *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

5. Because I have to pass this course in order to pass the module *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

6. Because I think programming is good for me *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

7. Because I have to follow the course *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

8. Because I think programming is pleasant *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

9. **There may be good reasons to learn OpenFrameworks, but personally I do not see any ***

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

10. **Because I believe being able to program in OpenFrameworks is a valuable skill in my skill set ***

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

11. **Because I am supposed to do it ***

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

12. **Because I feel that I have to do it ***

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

13. **Because it is fun to program ***

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

14. **But I am not sure OpenFrameworks is a good thing for me to pursue ***

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

15. **Because I believe it is important for me ***

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

16. **But I do not see why the course is given in OpenFrameworks ***

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

I'm learning how to Program with OpenCV...

17. **Because I am interested in OpenCV ***

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

18. But I am not sure if it is worth it *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

19. Because I feel good when working with OpenCV *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

20. Because I decided I want to learn about OpenCV *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

21. Because I have to pass this course in order to pass the module *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

22. Because I think being able to program with OpenCV is good for me *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

23. Because I have to follow the course *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

24. Because I think programming with OpenCV is pleasant *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

25. There may be good reasons to learn how to work with OpenCV, but personally I do not see any *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

26. Because I believe being able to program with OpenCV is a valuable skill in my skill set *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

27. Because I am supposed to do it *

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

28. **Because I feel that I have to do it ***

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

29. **Because it is fun to program with OpenCV ***

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

30. **But I am not sure if being able to program with OpenCV is a good thing for me to pursue ***

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

31. **Because I believe it is important for me ***

Mark only one oval.

	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

32. **But I do not see why the course teaches OpenCV ***

Mark only one oval.

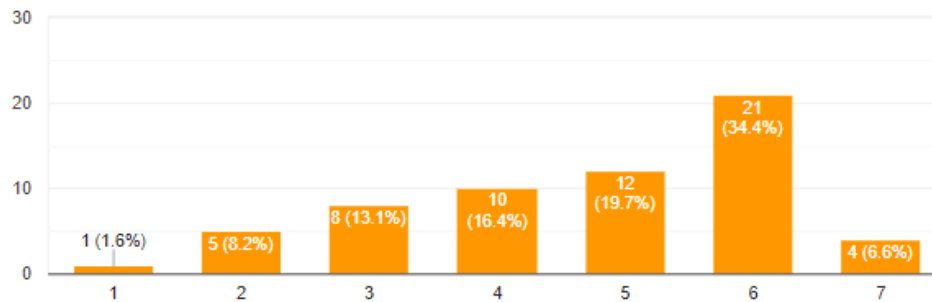
	1	2	3	4	5	6	7	
corresponds not at all	☐	☐	☐	☐	☐	☐	☐	corresponds exactly

Thank you very much!

H – RESULTS QUESTIONNAIRE 1

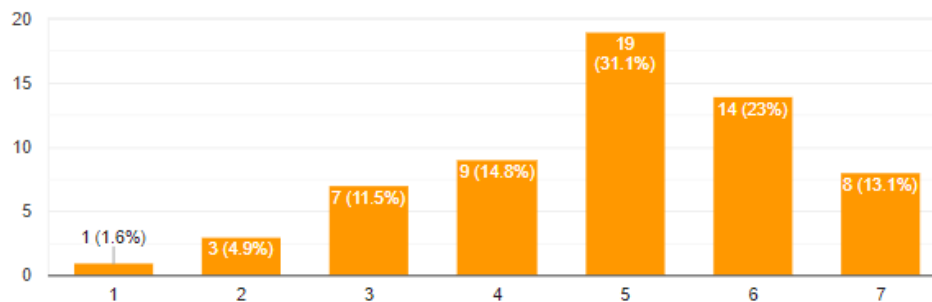
Because I am interested in programming

61 responses



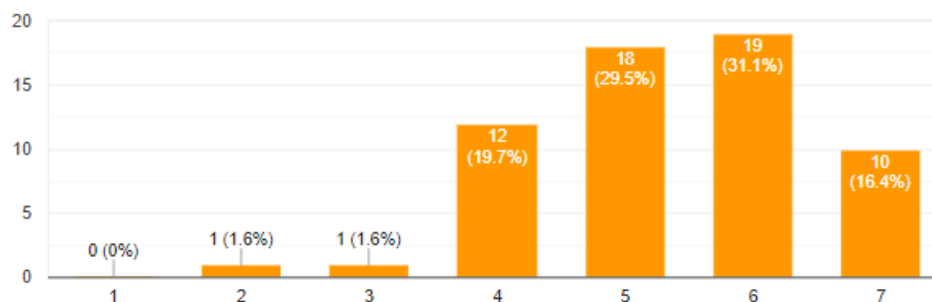
Because I believe being able to program in OpenFrameworks is a valuable skill in my skill set

61 responses



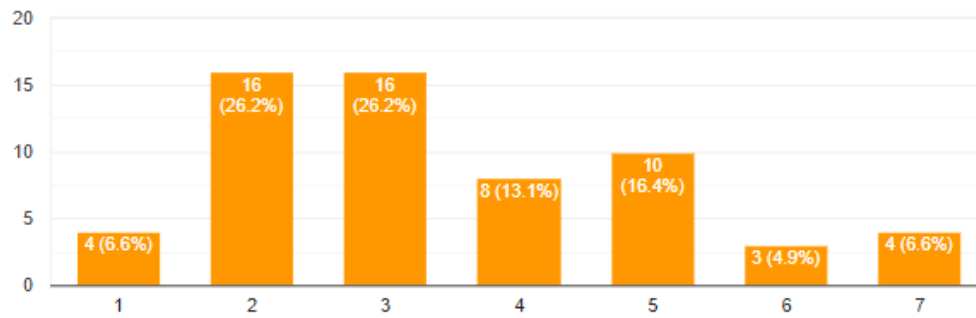
Because I am supposed to do it

61 responses



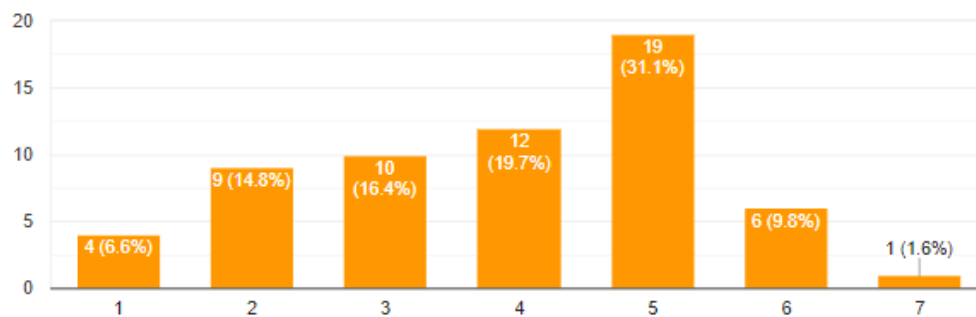
And there may be good reasons to learn OpenFrameworks, but personally I do not see any

61 responses



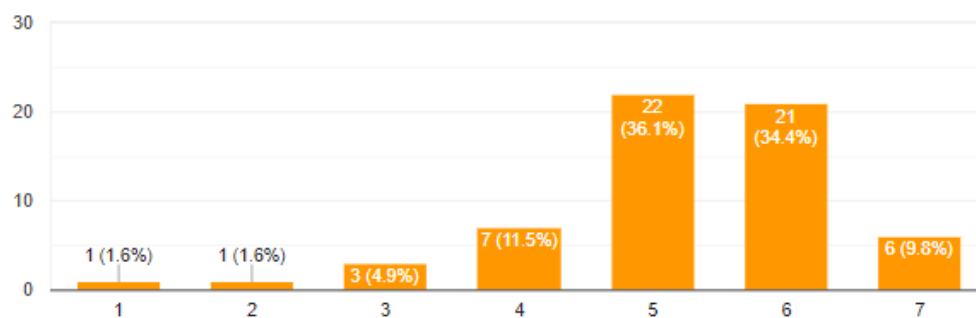
Because I think programming is pleasant

61 responses



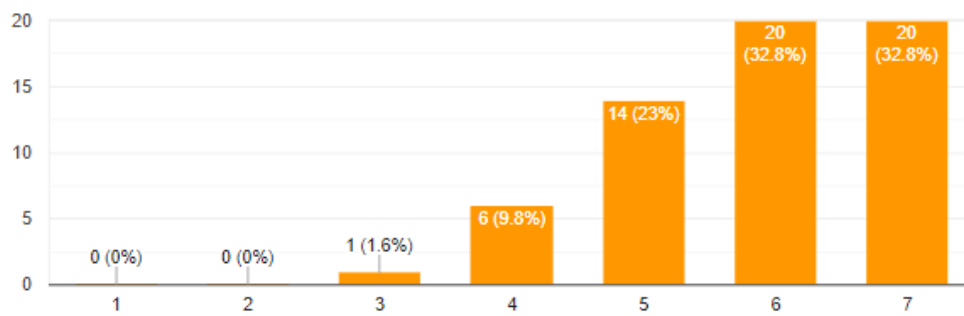
Because I think programming is good for me

61 responses



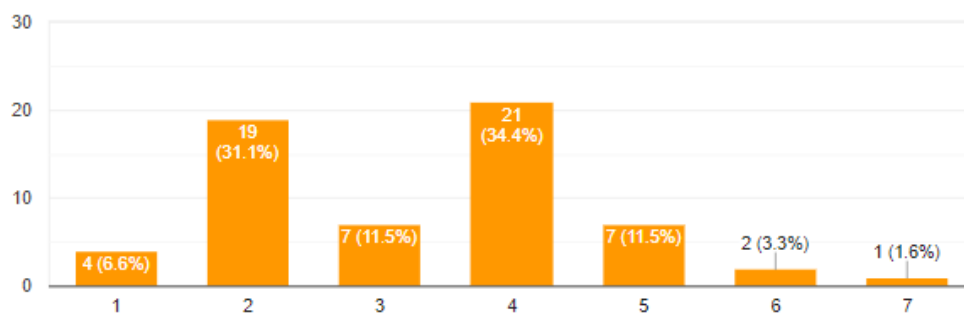
Because I have to pass this course in order to pass the module

61 responses



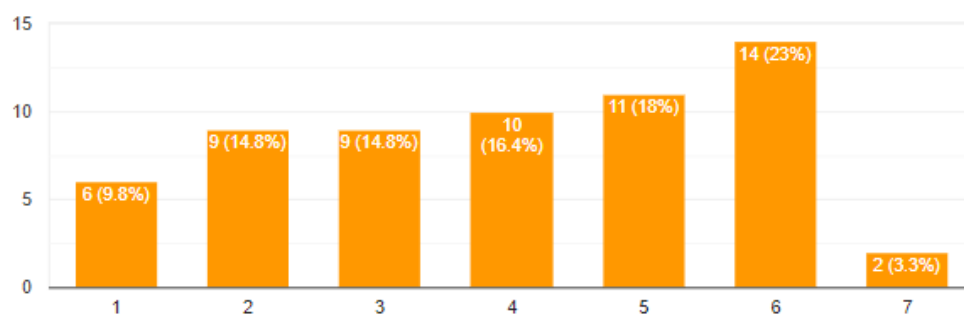
But I am not sure if it is worth it

61 responses



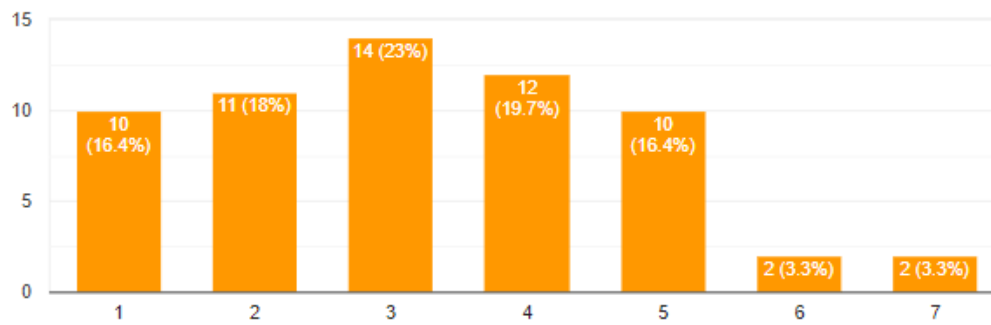
Because it is fun to program

61 responses



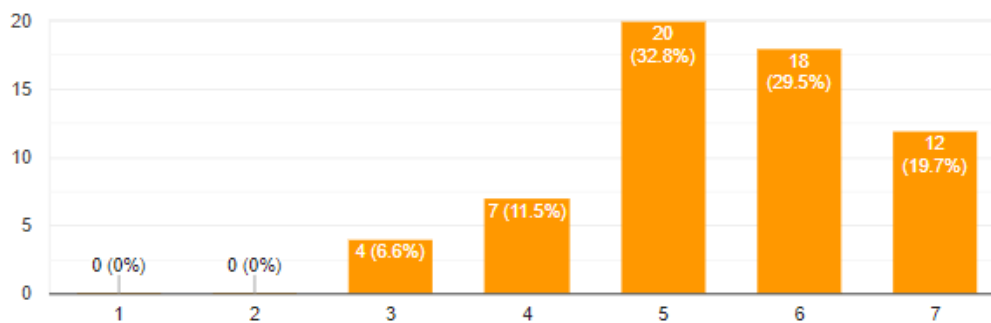
Because I decided I want to learn about OpenFrameworks

61 responses



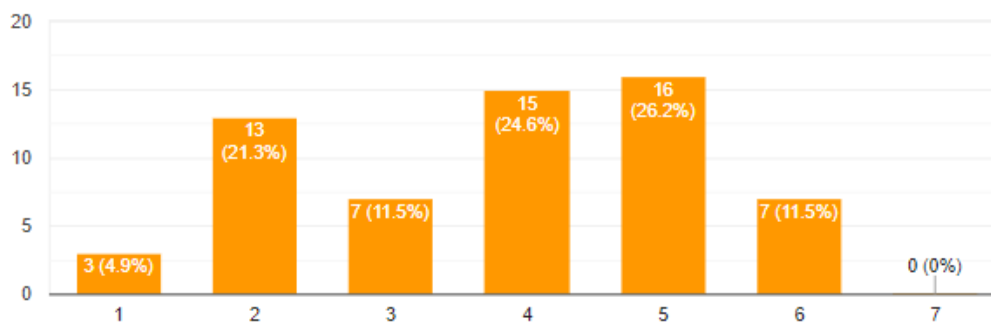
Because I have to follow the course

61 responses



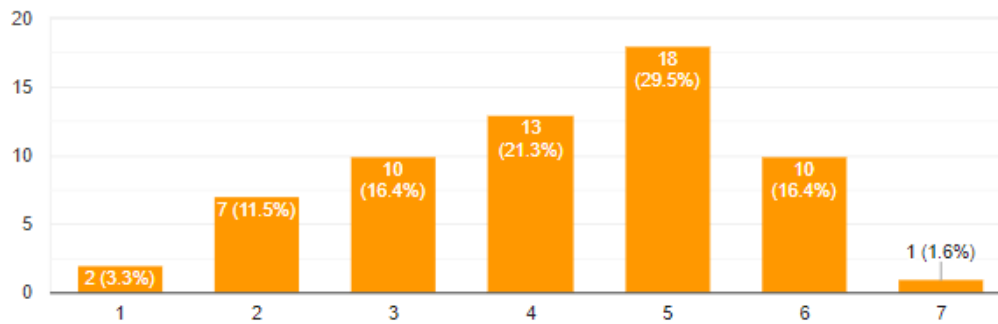
But I do not see why the course is given in OpenFrameworks

61 responses



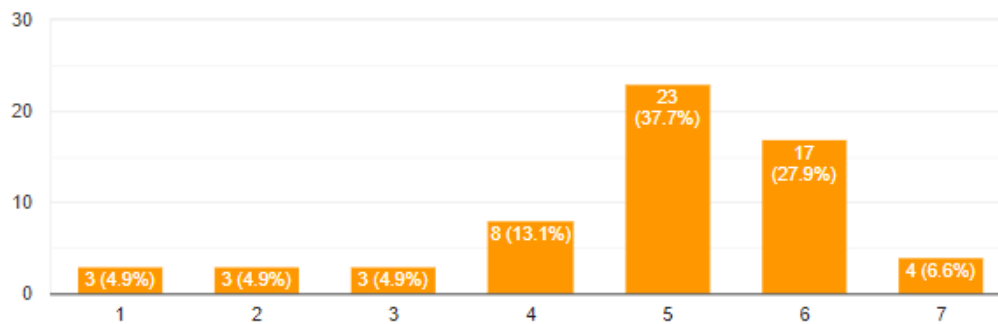
Because I feel good when doing a programming course

61 responses



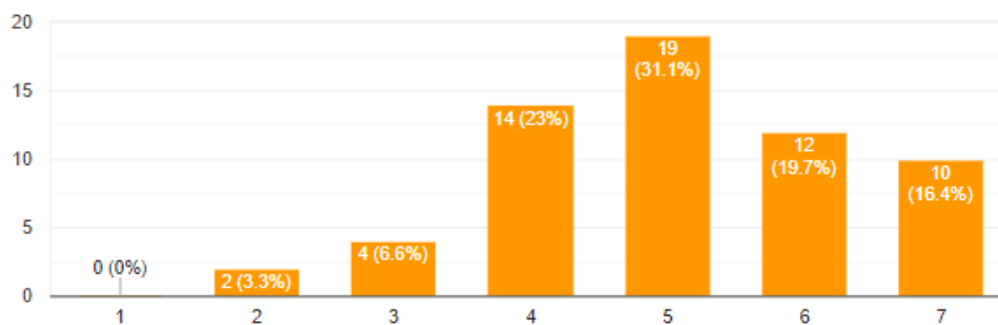
Because I believe it is important for me

61 responses



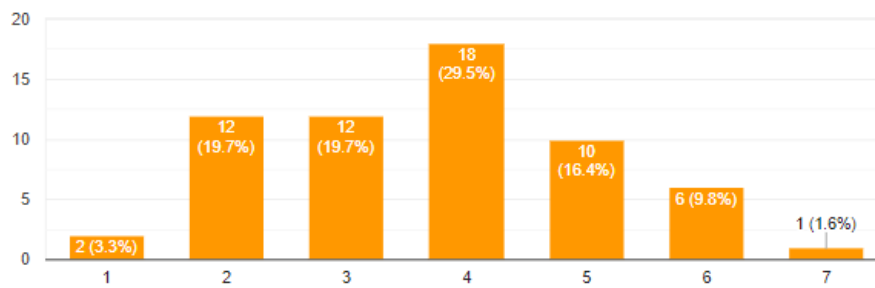
Because I feel that I have to do it

61 responses



But I am not sure OpenFrameworks is a good thing for me to pursue

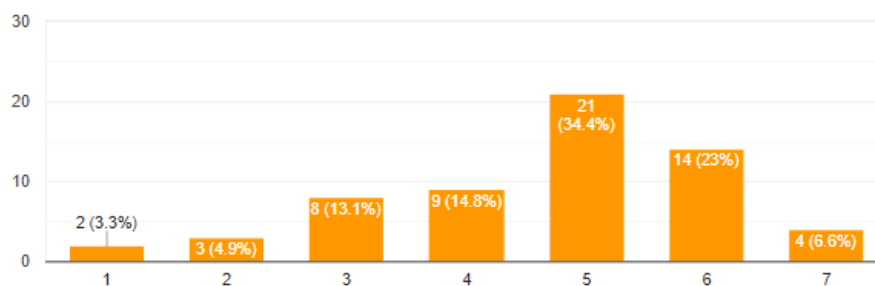
61 responses



I'm learning how to Program Search algorithms...

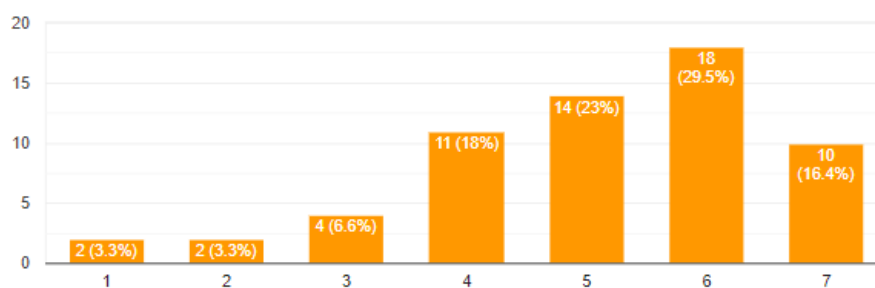
Because I am interested in Search algorithms

61 responses



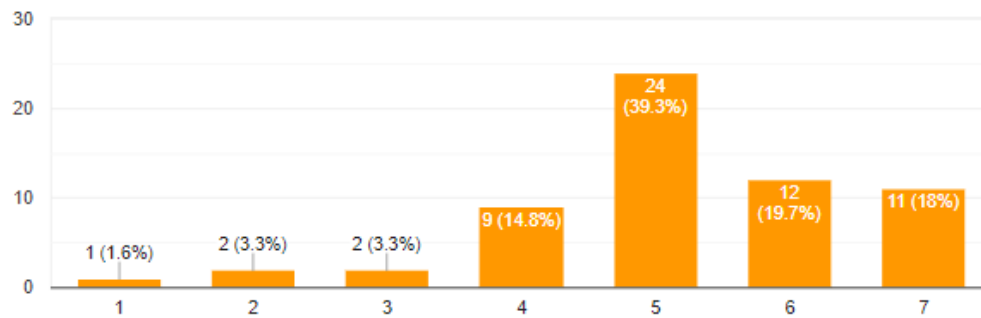
Because I believe being able to program with Search algorithms is a valuable skill in my skill set

61 responses



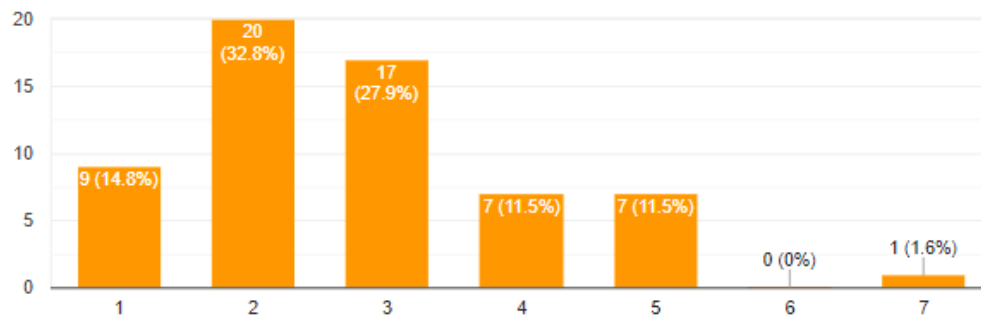
Because I am supposed to do it

61 responses



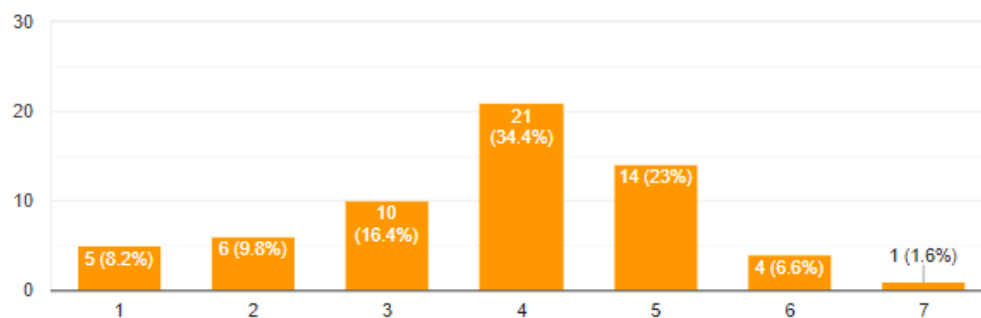
There may be good reasons to learn how to work with Search algorithms, but personally I do not see any

61 responses



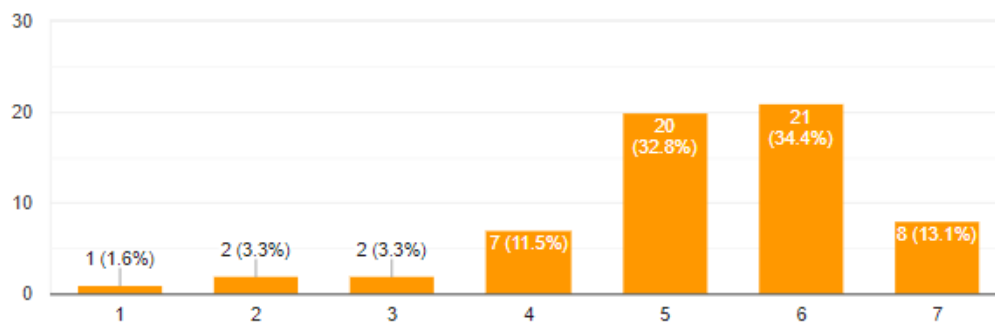
Because I think programming Search algorithms is pleasant

61 responses



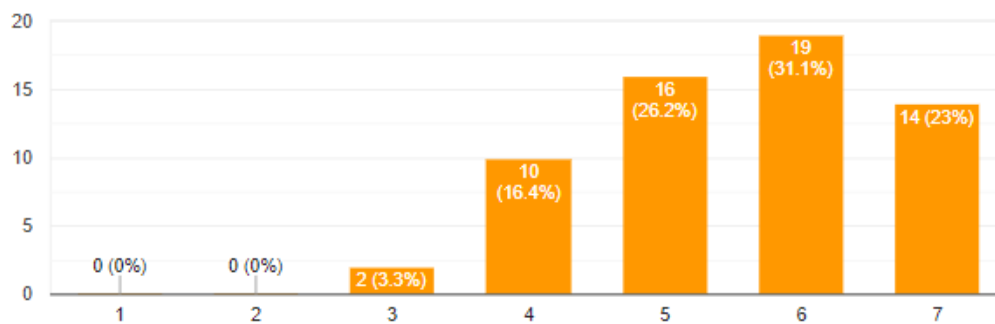
Because I think being able to program Search algorithms is good for me

61 responses



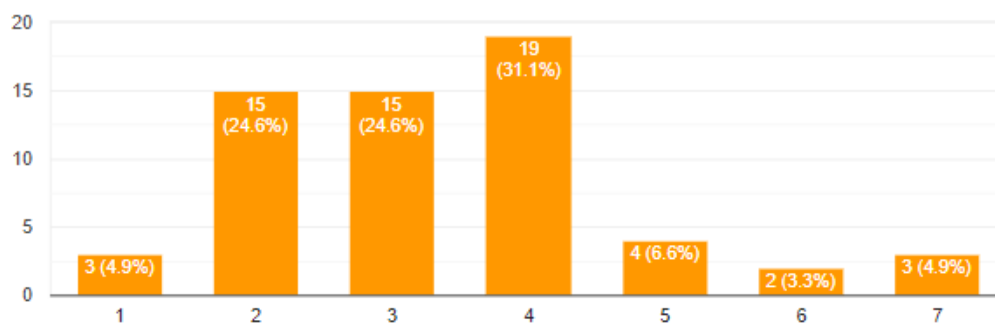
Because I have to pass this course in order to pass the module

61 responses



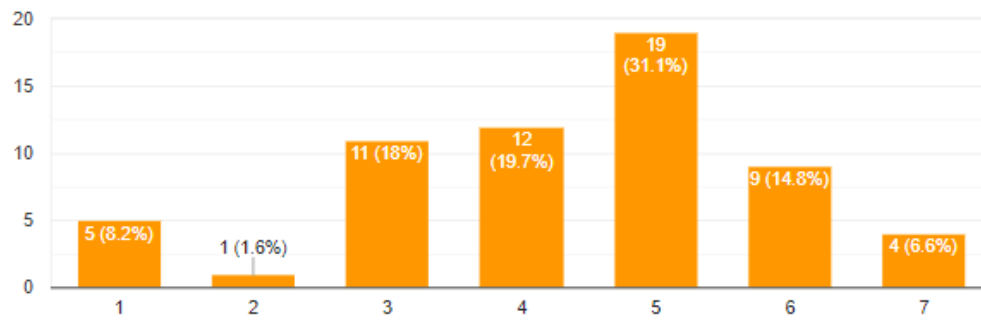
But I am not sure if it is worth it

61 responses



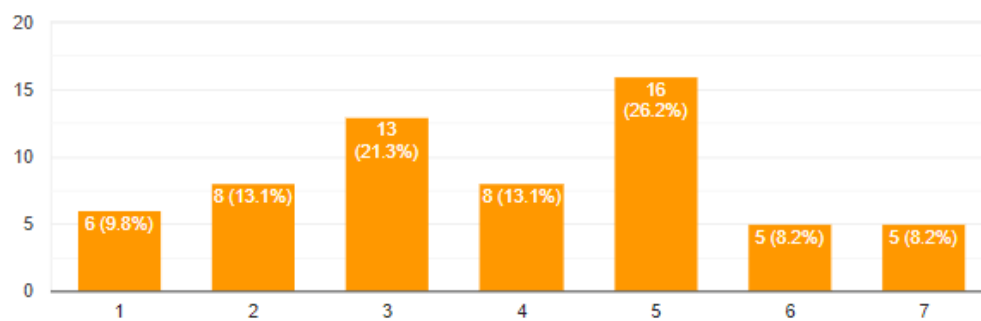
Because it is fun to program with Search algorithms

61 responses



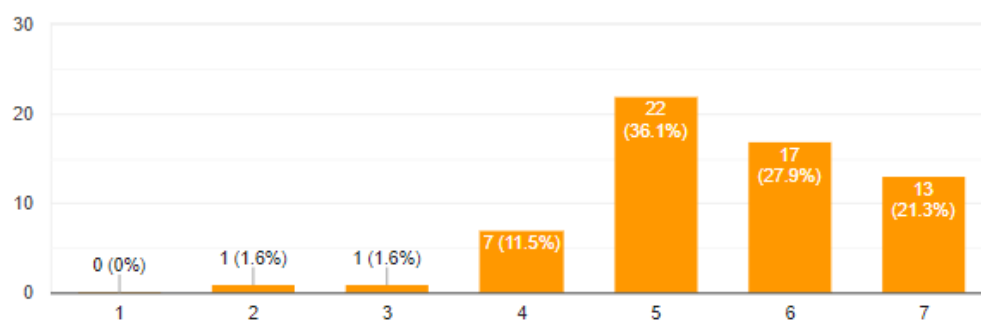
Because I decided I want to learn about Search algorithms

61 responses



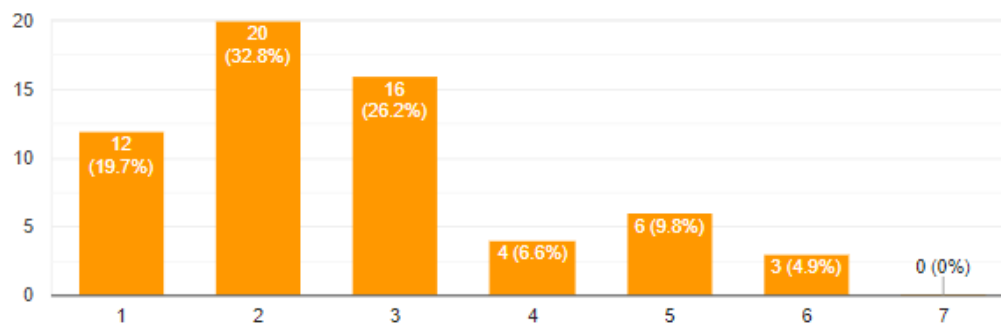
Because I have to follow the course

61 responses



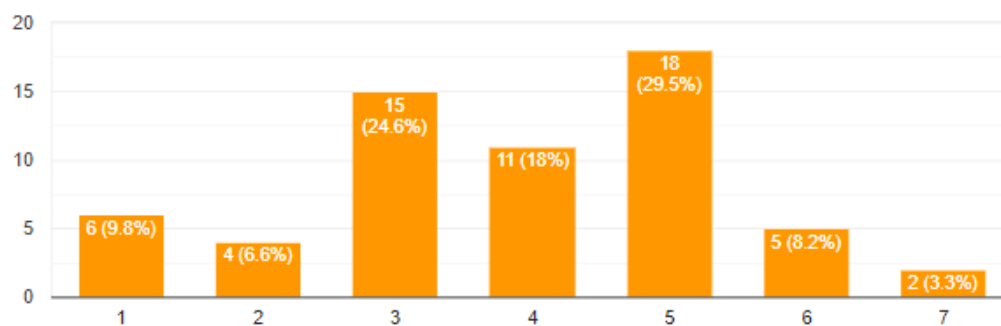
But I do not see why the course teaches Search algorithms

61 responses



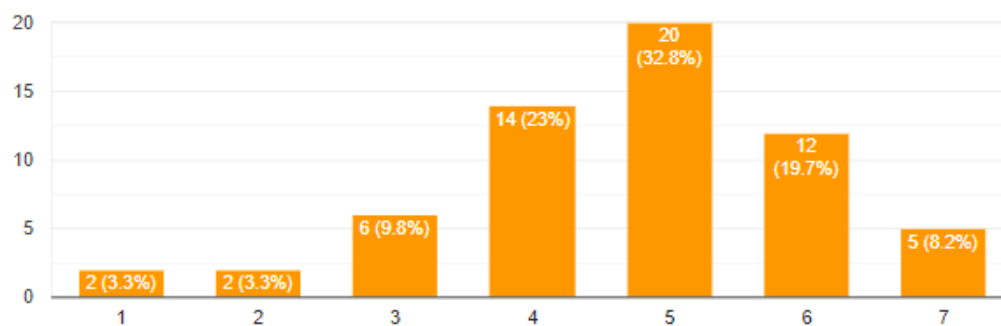
Because I feel good when working with Search algorithms

61 responses



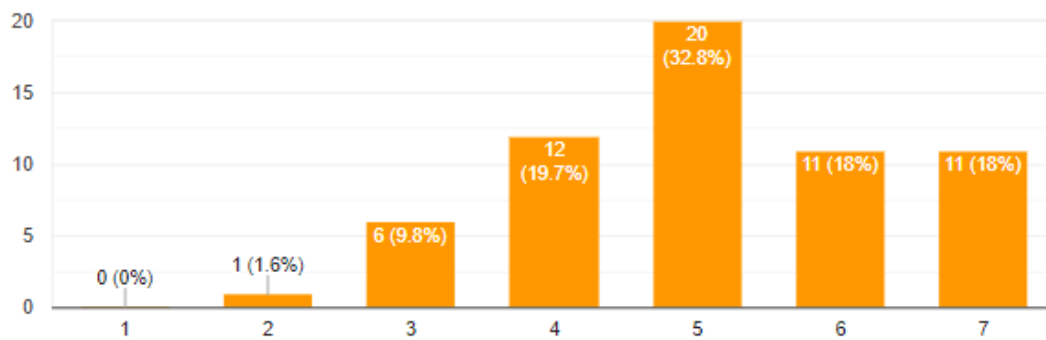
Because I believe it is important for me

61 responses



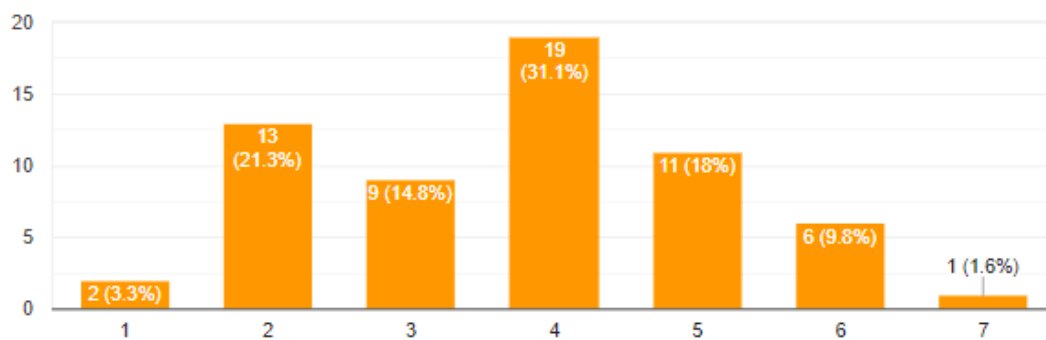
Because I feel that I have to do it

61 responses



But I am not sure if being able to program Search algorithms is a good thing for me to pursue

61 responses

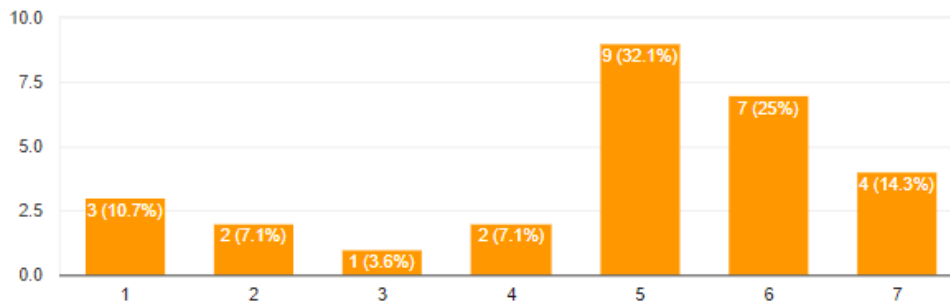


Thank you very much!

I – RESULTS QUESTIONNAIRE 2

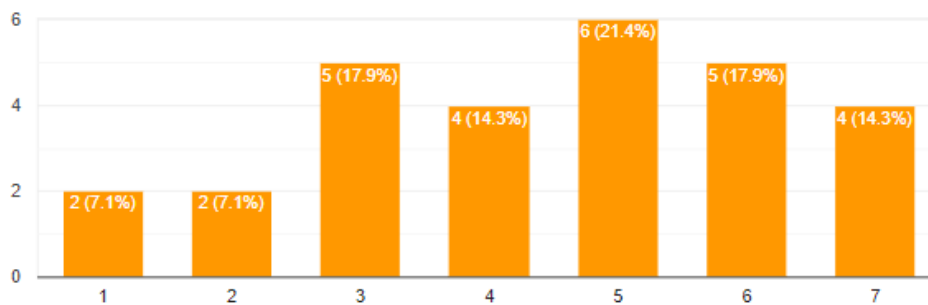
Because I am interested in programming

28 responses



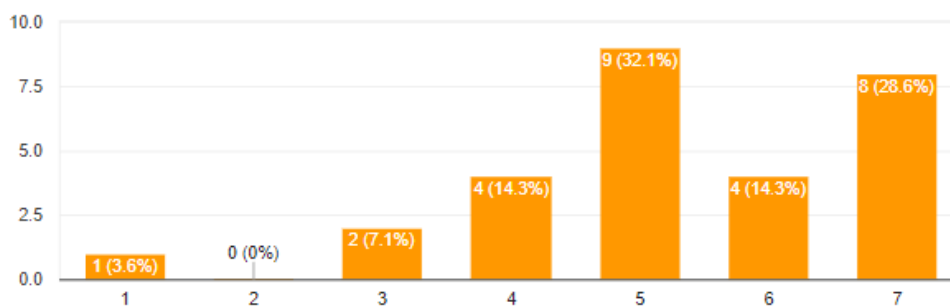
Because I believe being able to program in OpenFrameworks is a valuable skill in my skill set

28 responses



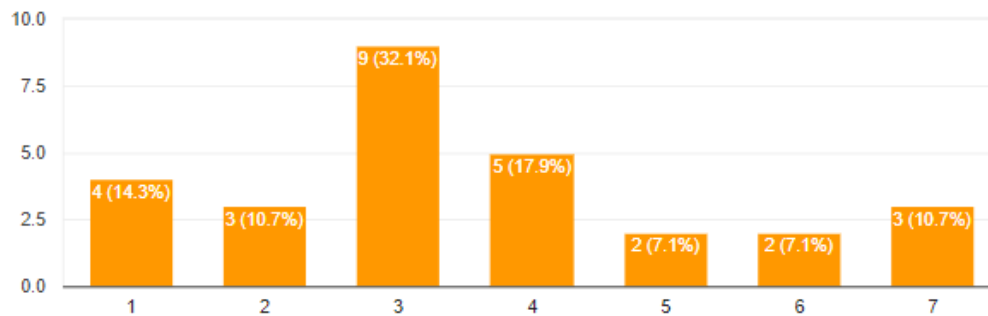
Because I am supposed to do it

28 responses



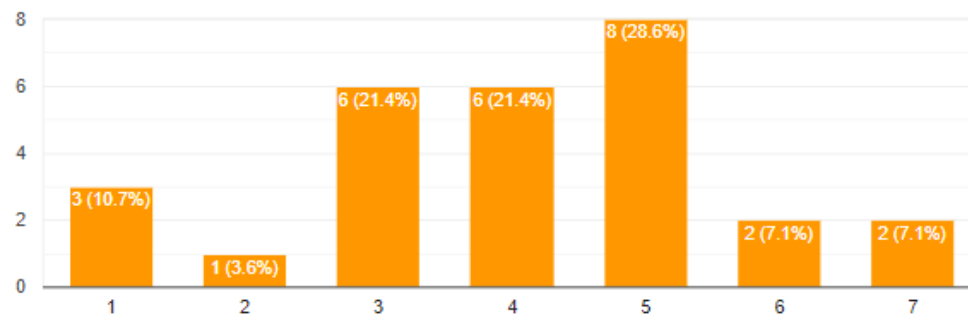
There may be good reasons to learn OpenFrameworks, but personally I do not see any

28 responses



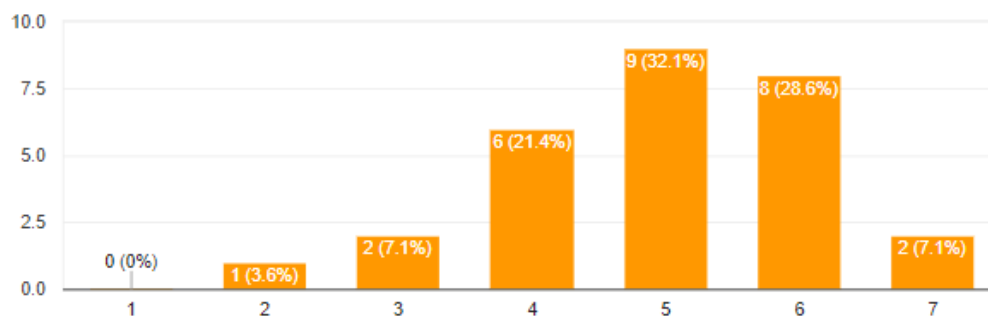
Because I think programming is pleasant

28 responses



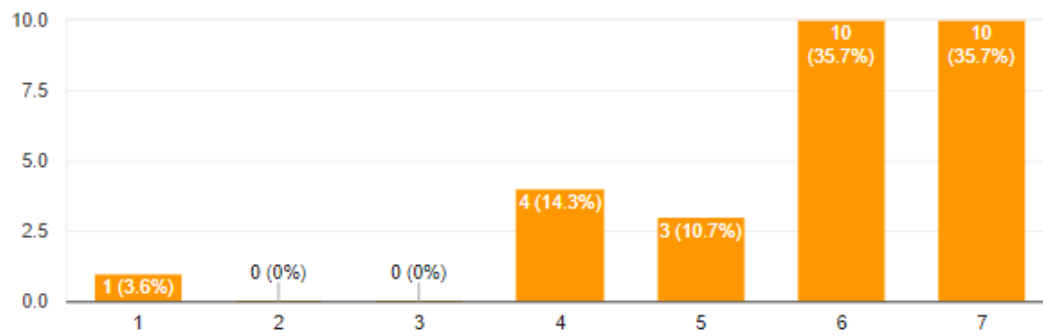
Because I think programming is good for me

28 responses



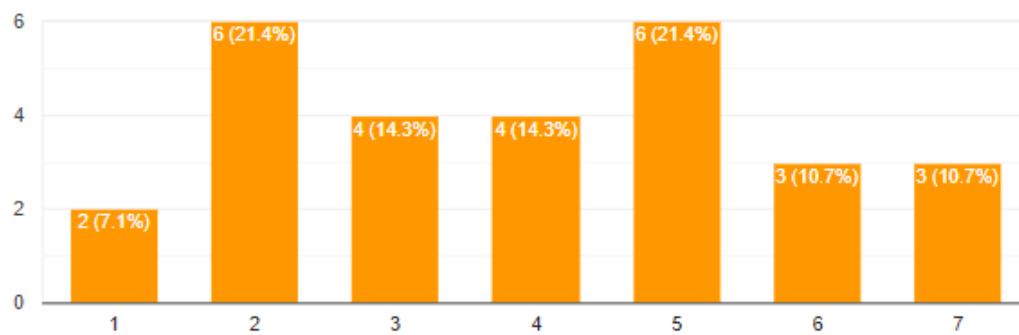
Because I have to pass this course in order to pass the module

28 responses



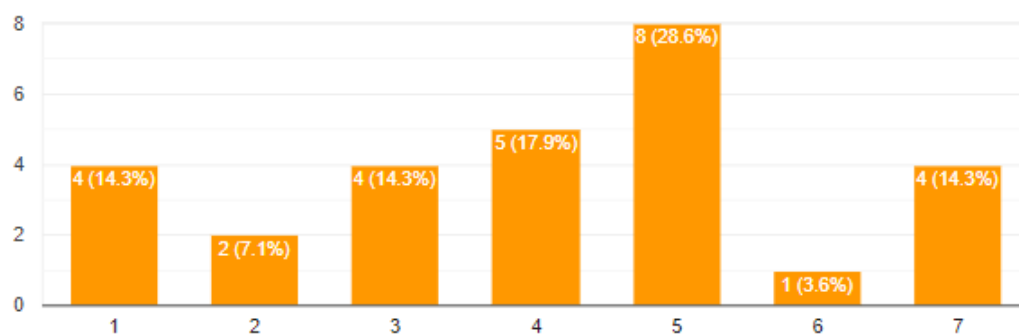
But I am not sure if it is worth it

28 responses



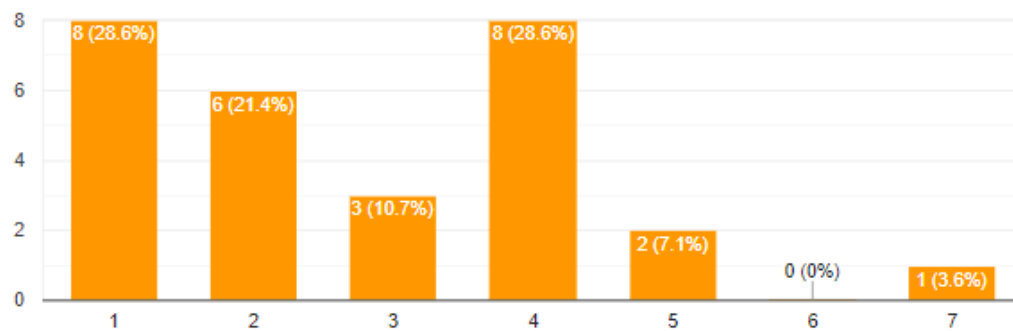
Because it is fun to program

28 responses



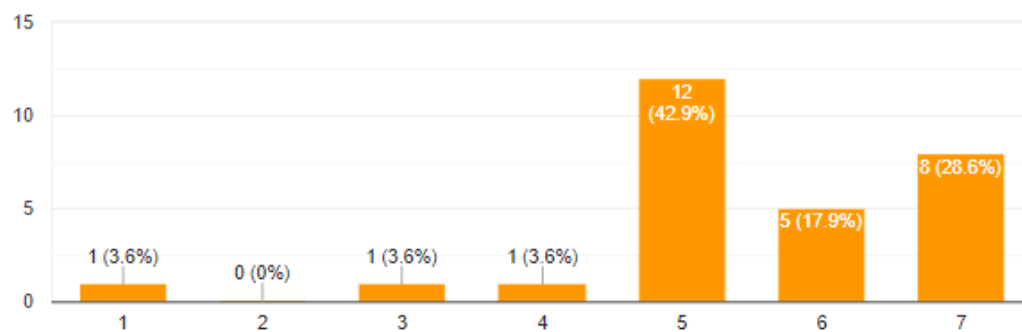
Because I decided I want to learn about OpenFrameworks

28 responses



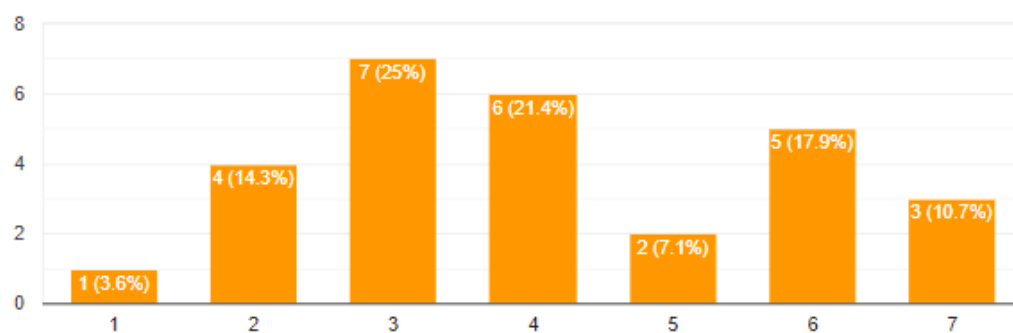
Because I have to follow the course

28 responses



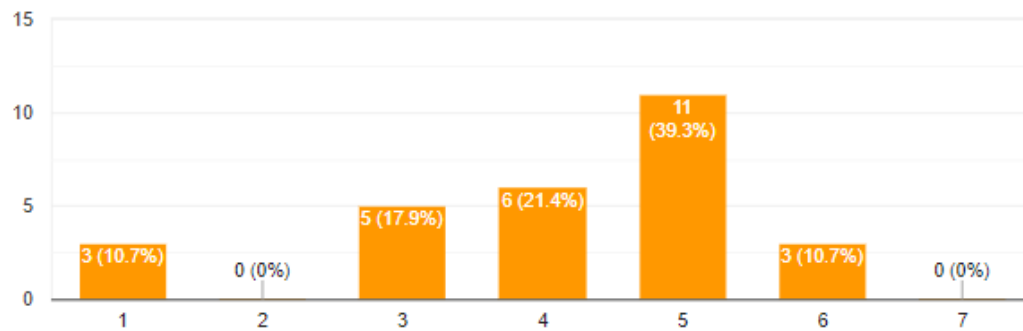
But I do not see why the course is given in OpenFrameworks

28 responses



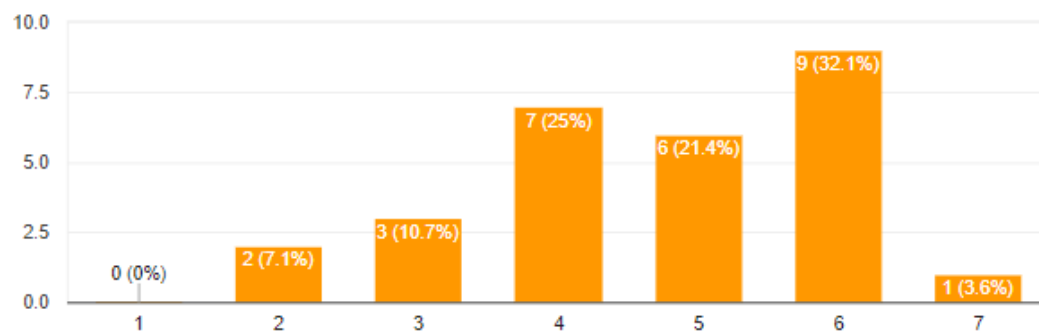
Because I feel good when doing a programming course

28 responses



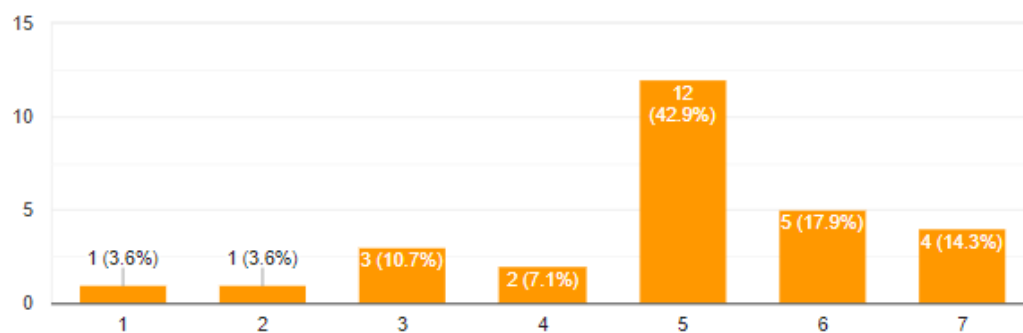
Because I believe it is important for me

28 responses



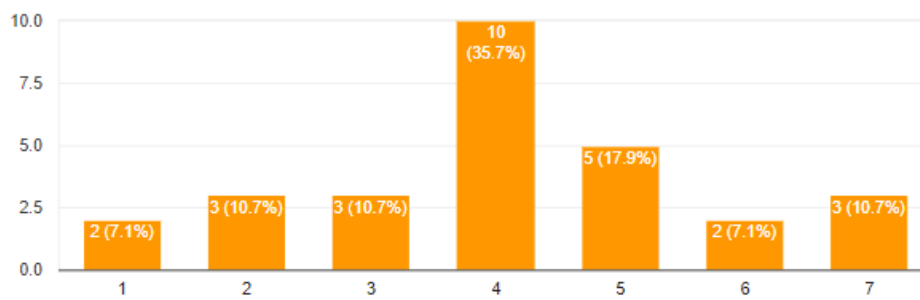
Because I feel that I have to do it

28 responses



But I am not sure OpenFrameworks is a good thing for me to pursue

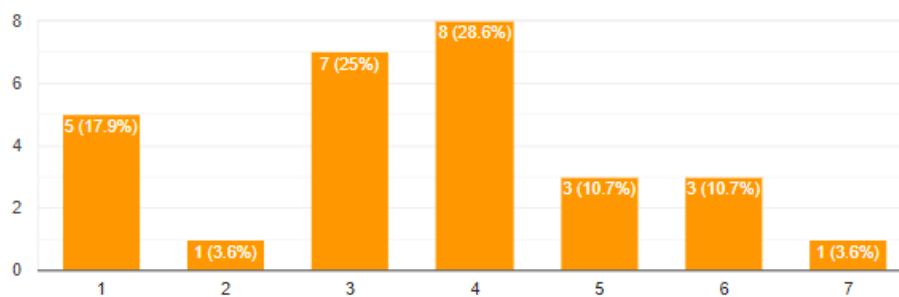
28 responses



I'm learning how to Program with OpenCV...

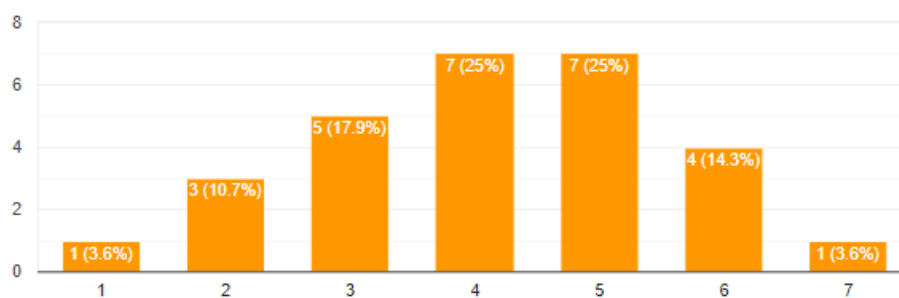
Because I am interested in OpenCV

28 responses



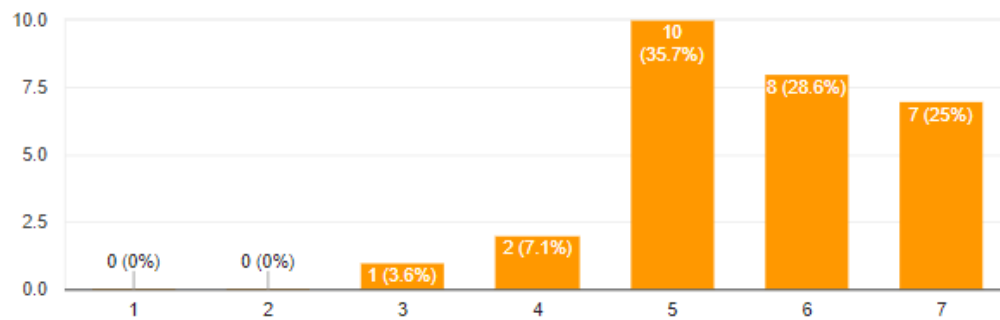
Because I believe being able to program with OpenCV is a valuable skill in my skill set

28 responses



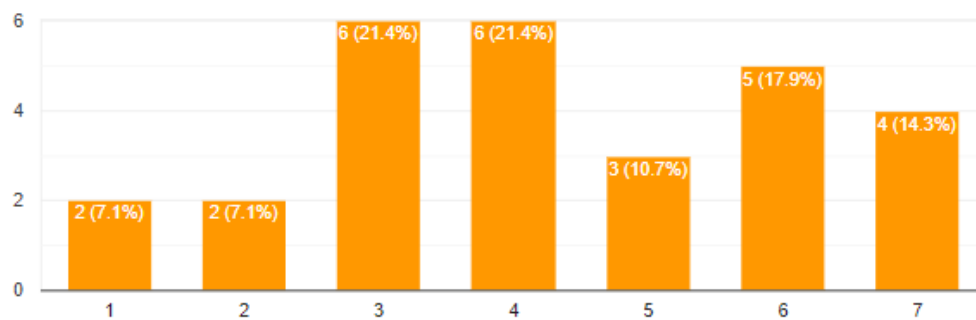
Because I am supposed to do it

28 responses



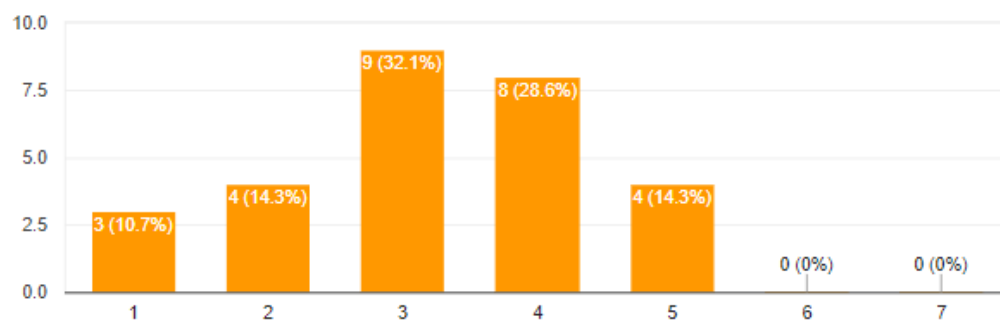
There may be good reasons to learn how to work with OpenCV, but personally I do not see any

28 responses



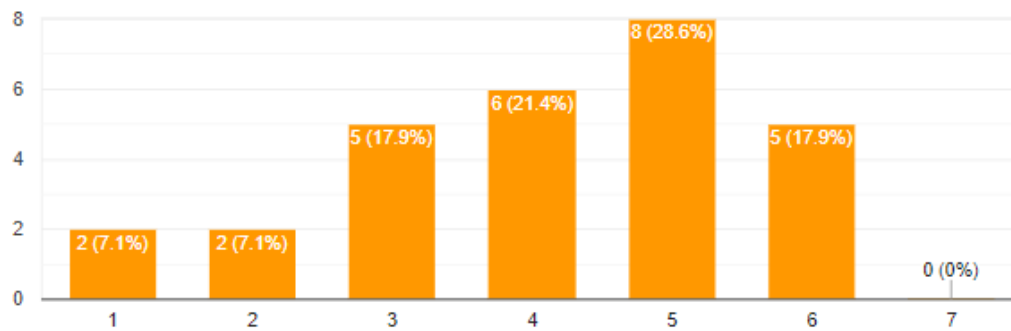
Because I think programming with OpenCV is pleasant

28 responses



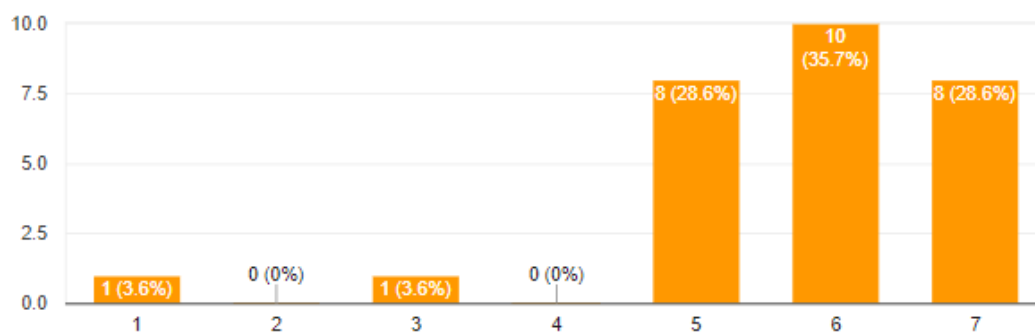
Because I think being able to program with OpenCV is good for me

28 responses



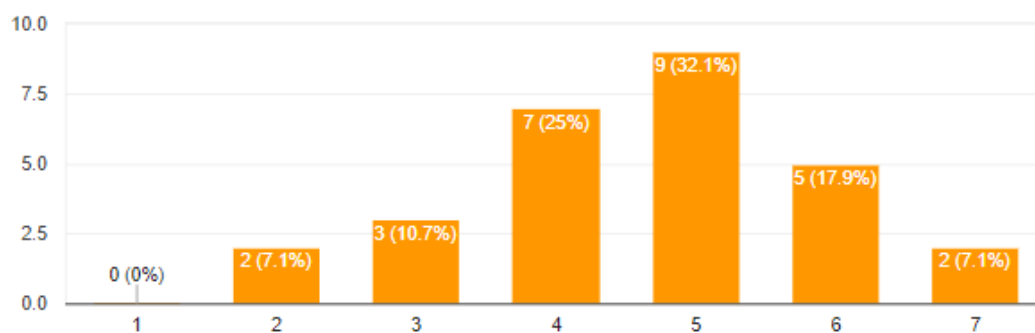
Because I have to pass this course in order to pass the module

28 responses



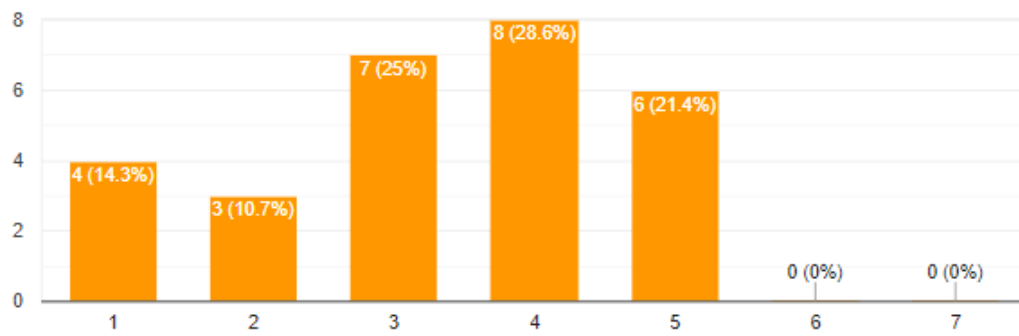
But I am not sure if it is worth it

28 responses



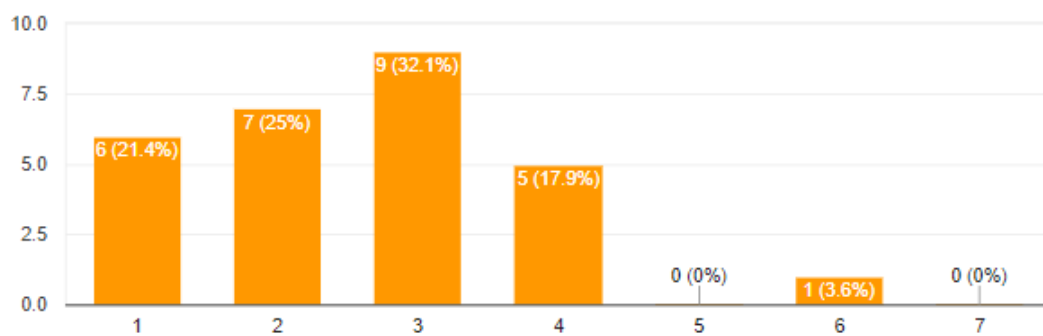
Because it is fun to program with OpenCV

28 responses



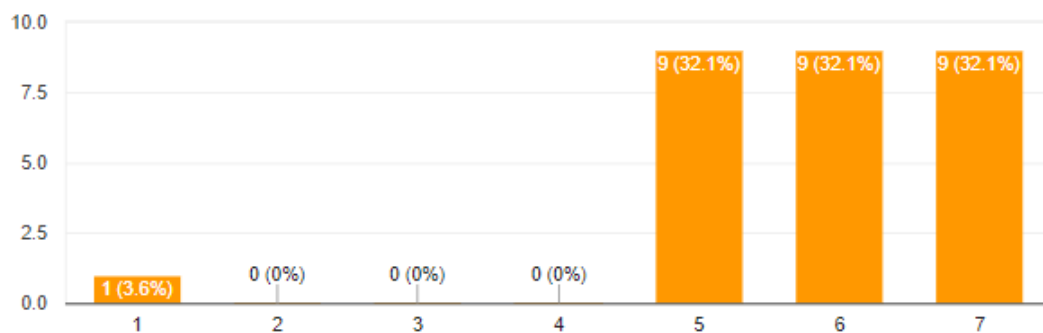
Because I decided I want to learn about OpenCV

28 responses



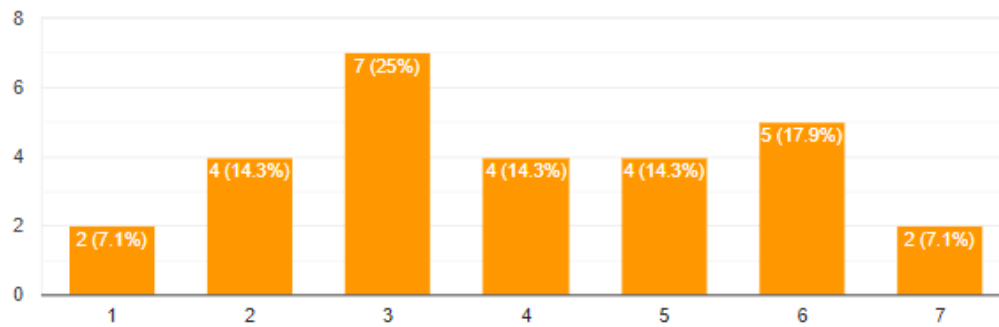
Because I have to follow the course

28 responses



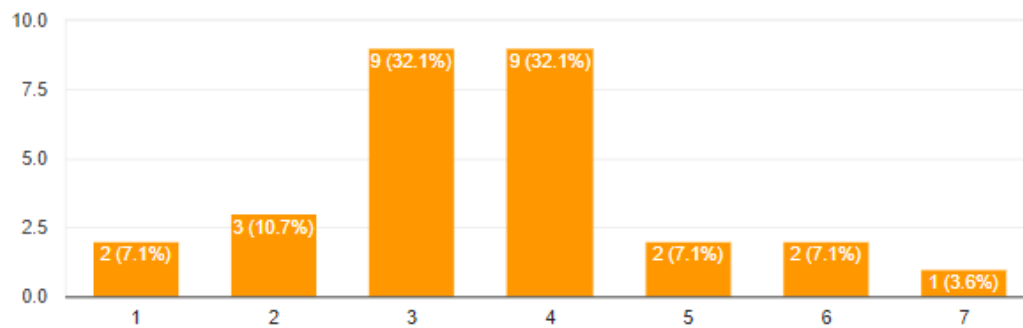
But I do not see why the course teaches OpenCV

28 responses



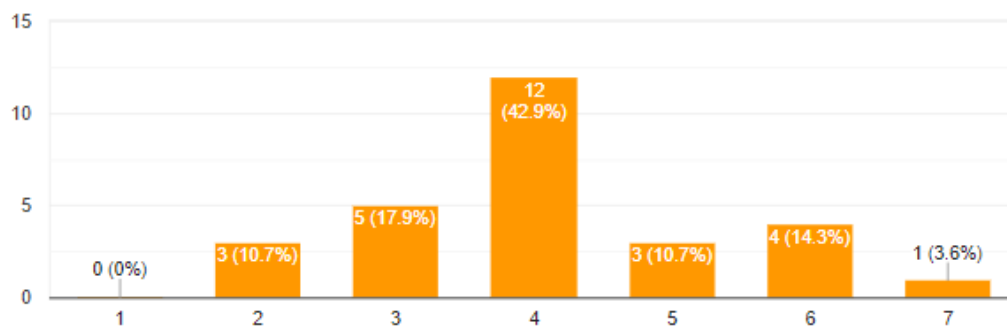
Because I feel good when working with OpenCV

28 responses



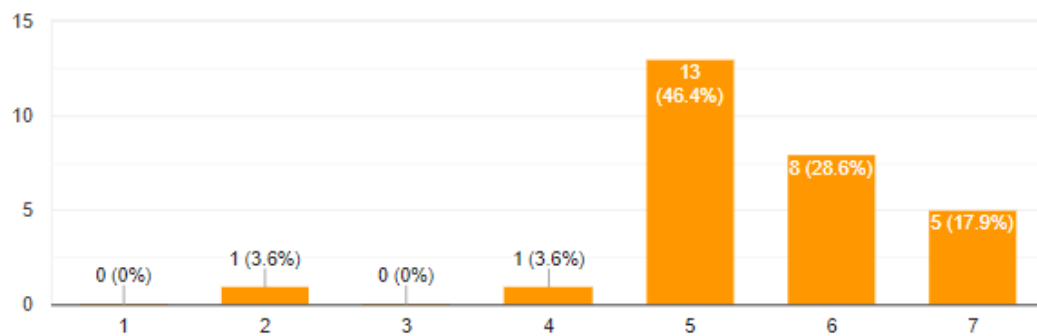
Because I believe it is important for me

28 responses



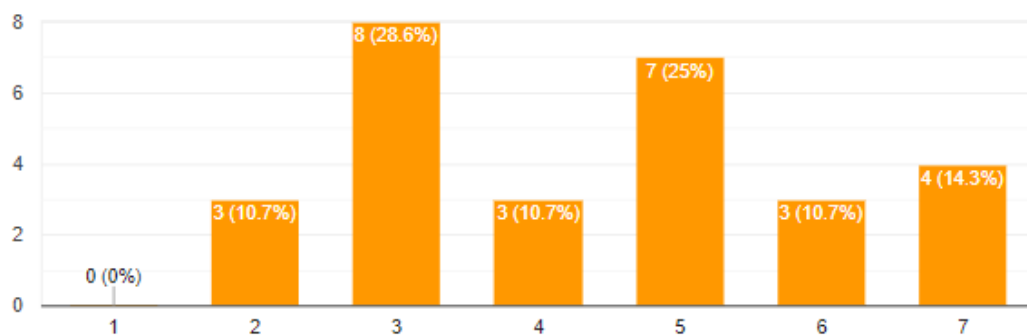
Because I feel that I have to do it

28 responses



But I am not sure if being able to program with OpenCV is a good thing for me to pursue

28 responses



Thank you very much!

J - STATISTICAL ANALYSIS

Normally (1978) 0.70 for Staphylococcus Resistant as Com.Bat & all & were higher than .7 most even $\geq .90$

Within my data set E1.1

the α for $I_m = .911$ with $m = 3.9 \leftrightarrow 4.7$ and $\theta = 1.42 \leftrightarrow 1.75$
 the α for $I_R = .841$ with $m = 3.2 \leftrightarrow 5.2$ and $\theta = 1.4 \leftrightarrow 1.58$
 the α for $E_R = .838$ with $m = 5.1 \leftrightarrow 5.9$ and $\theta = 1.05 \leftrightarrow 1.29$
 the α for $A_m = .800$ with $m = 3.3 \leftrightarrow 3.8$ and $\theta = 1.37 \leftrightarrow 1.62$

Relating to an overall correlation Std Scale!

the overall in are
 $I_m = 4.2$
 $I_R = 4.5 (5.0)$
 $E_R = 5.4$
 $A_m = 3.6$

within EI. 2

$$\alpha \quad I_m = .9_{.15}$$

$$I_R = .853$$

$$(.905)$$

$$\alpha \quad ER = .900$$

$$A_m = .723$$

$$m = 3.8 \leftrightarrow 4.7$$

$$m = 3.9 \leftrightarrow 5.3$$

$$(4.7 \leftrightarrow 5.3)$$

$$m = 5.1 \leftrightarrow 5.5$$

$$m = 2.7 \leftrightarrow 3.8$$

$$\theta = 1.35 \leftrightarrow 1.55$$

$$\theta = 1.28 \leftrightarrow 1.50$$

$$(1.28 \leftrightarrow 1.50)$$

$$\theta 1.16 \leftrightarrow 1.32$$

$$\theta 1.32 \leftrightarrow 1.41$$

$$\text{Overall } \vec{m} \quad I_m = 4.2 \mid I_R = 4.7 \mid ER = 5.3 \mid A_m = 3.2$$

Combined (EI. 1 & 2)

$$\alpha \quad m$$

$$I_m \quad .938 \quad 3.8 \quad 4.7$$

$$I_R \quad .902$$

$$(.900)$$

$$ER \quad .916$$

$$A_m \quad .849$$

$$\theta$$

$$1.389 - 1.75$$

$$\vec{m}$$

$$4.191$$

$$4.6$$

$$(4.8)$$

$$5.4$$

$$3.4$$

E2.1	α	m	θ	$\vec{m}$		
I _m	.947	4.0	4.7	1.42	1.88	4.24
I _R	.640	2.8	5.0	1.20	1.79	4.23
*	(.685)	(4.5)	5.0	(1.20)	(1.79)	(4.71)
E _R	.894	4.9	5.8	1.37	1.51	5.4
A _m	.809	3.5	4.1	1.64	1.84	3.9

E2.2	α	m	θ	$\vec{m}$		
I _m	.759	3.2	3.6	1.20	1.67	3.43
I _R	.735 (.734)	2.6 (4.1)	4.1	1.23 (1.28)	1.47	3.7 (4.1)
E _R	.816	5.5	5.8	1.08	1.33	5.7
A _m	.781	4.0	4.6	1.31	1.81	4.3

2

Combined E2. 1&2	
I _m	α $\vec{m}$
I _R	.797 (.805) 4.0 (4.2)
E _R	.919 5.5
A _m	.826 4.1

Measurement 1

General Course

NEW FILE.

DATASET NAME DataSet1 WINDOW=FRONT.

RELIABILITY

/VARIABLES=VAR00001 VAR00005 VAR00009 VAR00013

/SCALE('ALL VARIABLES') ALL

/MODEL=ALPHA

/STATISTICS=DESCRIPTIVE SCALE CORR

/SUMMARY=TOTAL MEANS VARIANCE

Reliability

[DataSet1]

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	61	100.0
	Excluded ^a	0	.0
	Total	61	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.911	.912	4

Item Statistics

	Mean	Std. Deviation	N
VAR00001	4.7377	1.50446	61
VAR00005	3.9016	1.49114	61
VAR00009	4.0000	1.75119	61
VAR00013	4.1803	1.42019	61

Inter-Item Correlation Matrix

	VAR00001	VAR00005	VAR00009	VAR00013
VAR00001	1.000	.679	.728	.709
VAR00005	.679	1.000	.804	.693
VAR00009	.728	.804	1.000	.717
VAR00013	.709	.693	.717	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	4.205	3.902	4.738	.836	1.214	.139
Item Variances	2.393	2.017	3.067	1.050	1.520	.214

Summary Item Statistics

	N of Items
Item Means	4
Item Variances	4

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00001	12.0820	18.010	.776	.609	.891
VAR00005	12.9180	17.777	.810	.681	.880
VAR00009	12.8197	15.517	.841	.723	.871
VAR00013	12.6393	18.634	.777	.607	.892

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
16.8197	30.184	5.49396	4

RELIABILITY

```

/VARIABLES=VAR00002 VAR00006 VAR00010 VAR00014
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	61	100.0
	Excluded ^a	0	.0
	Total	61	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.813	.817	4

VS .793

Item Statistics

	Mean	Std. Deviation	N
VAR00002	4.9016	1.44574	61
VAR00006	5.2131	1.19882	61
VAR00010	3.2459	1.57751	61
VAR00014	4.8361	1.45121	61

Inter-Item Correlation Matrix

	VAR00002	VAR00006	VAR00010	VAR00014
VAR00002	1.000	.378	.435	.620
VAR00006	.378	1.000	.465	.691
VAR00010	.435	.465	1.000	.571
VAR00014	.620	.691	.571	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	4.549	3.246	5.213	1.967	1.606	.782
Item Variances	2.030	1.437	2.489	1.051	1.732	.190

Summary Item Statistics

	N of Items
Item Means	4
Item Variances	4

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00002	13.2951	12.811	.569	.400	.794
VAR00006	12.9836	13.883	.613	.491	.778
VAR00010	14.9508	11.948	.583	.348	.793
VAR00014	13.3607	11.068	.789	.656	.685

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
18.1967	20.794	4.56004	4

RELIABILITY

```

/VARIABLES=VAR00003 VAR00007 VAR00011 VAR00015
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	61	100.0
	Excluded ^a	0	.0
	Total	61	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.838	.837	4

Item Statistics

	Mean	Std. Deviation	N
VAR00003	5.3607	1.12595	61
VAR00007	5.8525	1.04620	61
VAR00011	5.4426	1.13320	61
VAR00015	5.0656	1.28930	61

Inter-Item Correlation Matrix

	VAR00003	VAR00007	VAR00011	VAR00015
VAR00003	1.000	.470	.461	.707
VAR00007	.470	1.000	.506	.613
VAR00011	.461	.506	1.000	.619
VAR00015	.707	.613	.619	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	5.430	5.066	5.852	.787	1.155	.105
Item Variances	1.327	1.095	1.662	.568	1.519	.057

Summary Item Statistics

	N of Items
Item Means	4
Item Variances	4

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00003	16.3607	8.701	.653	.502	.803
VAR00007	15.8689	9.249	.622	.404	.817
VAR00011	16.2787	8.838	.621	.409	.817
VAR00015	16.6557	7.130	.801	.653	.733

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
21.7213	14.304	3.78211	4

RELIABILITY

/VARIABLES=VAR00004 VAR00008 VAR00012 VAR00016

```

/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	61	100.0
	Excluded ^a	0	.0
	Total	61	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.800	.805	4

Item Statistics

	Mean	Std. Deviation	N
VAR00004	3.4754	1.61871	61
VAR00008	3.2951	1.37045	61
VAR00012	3.8033	1.45853	61
VAR00016	3.7213	1.39201	61

Inter-Item Correlation Matrix

	VAR00004	VAR00008	VAR00012	VAR00016
VAR00004	1.000	.462	.421	.526
VAR00008	.462	1.000	.405	.717
VAR00012	.421	.405	1.000	.514
VAR00016	.526	.717	.514	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	3.574	3.295	3.803	.508	1.154	.054
Item Variances	2.141	1.878	2.620	.742	1.395	.113

Summary Item Statistics

	N of Items
Item Means	4
Item Variances	4

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00004	10.8197	12.384	.562	.320	.780
VAR00008	11.0000	13.133	.644	.524	.736
VAR00012	10.4918	13.587	.530	.297	.789
VAR00016	10.5738	12.282	.737	.598	.691

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
14.2951	21.411	4.62725	4

RELIABILITY

```

/VARIABLES=VAR00002 VAR00006 VAR00014
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	61	100.0
	Excluded ^a	0	.0
	Total	61	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.793	.794	3

✓ 5.813

Item Statistics

	Mean	Std. Deviation	N
VAR00002	4.9016	1.44574	61
VAR00006	5.2131	1.19882	61
VAR00014	4.8361	1.45121	61

Inter-Item Correlation Matrix

	VAR00002	VAR00006	VAR00014
VAR00002	1.000	.378	.620
VAR00006	.378	1.000	.691
VAR00014	.620	.691	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	4.984	4.836	5.213	.377	1.078	.041
Item Variances	1.878	1.437	2.106	.669	1.465	.146

Summary Item Statistics

	N of Items
Item Means	3
Item Variances	3

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item-Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00002	10.0492	5.948	.554	.389	.809
VAR00006	9.7377	6.797	.594	.482	.765
VAR00014	10.1148	4.837	.784	.628	.541

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
14.9508	11.948	3.45652	3

Measurement 1 Search

RELIABILITY

```
/VARIABLES=VAR00001 VAR00005 VAR00009 VAR00013
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE
```

Reliability

[DataSet1]

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	61	100.0
	Excluded ^a	0	.0
	Total	61	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.915	.916	4

Item Statistics

	Mean	Std. Deviation	N
VAR00001	4.6721	1.43436	61
VAR00005	3.8033	1.38828	61
VAR00009	4.3443	1.54796	61
VAR00013	3.8852	1.52860	61

Inter-Item Correlation Matrix

	VAR00001	VAR00005	VAR00009	VAR00013
VAR00001	1.000	.729	.660	.689
VAR00005	.729	1.000	.815	.775
VAR00009	.660	.815	1.000	.728
VAR00013	.689	.775	.728	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	4.176	3.803	4.672	.869	1.228	.166
Item Variances	2.179	1.927	2.396	.469	1.243	.050

Summary Item Statistics

	N of Items
Item Means	4
Item Variances	4

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00001	12.0328	16.899	.751	.574	.909
VAR00005	12.9016	16.190	.868	.760	.871
VAR00009	12.3607	15.534	.810	.691	.890
VAR00013	12.8197	15.717	.805	.653	.891

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
16.7049	27.811	5.27366	4

RELIABILITY

```

/VARIABLES=VAR00002 VAR00006 VAR00010 VAR00014
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	61	100.0
	Excluded ^a	0	.0
	Total	61	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.859	.872	4

Item Statistics

	Mean	Std. Deviation	N
VAR00002	5.0820	1.49772	61
VAR00006	5.2623	1.27674	61
VAR00010	3.9016	1.72921	61
VAR00014	4.7049	1.38256	61

Inter-Item Correlation Matrix

	VAR00002	VAR00006	VAR00010	VAR00014
VAR00002	1.000	.773	.441	.752
VAR00006	.773	1.000	.495	.772
VAR00010	.441	.495	1.000	.545
VAR00014	.752	.772	.545	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	4.738	3.902	5.262	1.361	1.349	.365
Item Variances	2.194	1.630	2.990	1.360	1.834	.345

Summary Item Statistics

	N of Items
Item Means	4
Item Variances	4

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00002	13.8689	14.049	.744	.658	.803
VAR00006	13.6885	15.151	.791	.686	.793
VAR00010	15.0492	14.581	.536	.311	.905
VAR00014	14.2459	14.289	.808	.682	.779

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
18.9508	24.648	4.96463	4

RELIABILITY

```

/VARIABLES=VAR00002 VAR00006 VAR00014
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	61	100.0
	Excluded ^a	0	.0
	Total	61	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.905	.907	3

Item Statistics

	Mean	Std. Deviation	N
VAR00002	5.0820	1.49772	61
VAR00006	5.2623	1.27674	61
VAR00014	4.7049	1.38256	61

Inter-Item Correlation Matrix

	VAR00002	VAR00006	VAR00014
VAR00002	1.000	.773	.752
VAR00006	.773	1.000	.772
VAR00014	.752	.772	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	5.016	4.705	5.262	.557	1.118	.081
Item Variances	1.928	1.630	2.243	.613	1.376	.094

Summary Item Statistics

	N of Items
Item Means	3
Item Variances	3

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00002	9.9672	6.266	.810	.658	.870
VAR00006	9.7869	7.270	.825	.681	.857
VAR00014	10.3443	6.830	.808	.656	.866

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
15.0492	14.581	3.81849	3

RELIABILITY

```
/VARIABLES=VAR00003 VAR00007 VAR00011 VAR00015  
/SCALE('ALL VARIABLES') ALL  
/MODEL=ALPHA  
/STATISTICS=DESCRIPTIVE SCALE CORR
```

/SUMMARY=TOTAL MEANS VARIANCE.

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	61	100.0
	Excluded ^a	0	.0
	Total	61	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.900	.903	4

Item Statistics

	Mean	Std. Deviation	N
VAR00003	5.1803	1.32298	61
VAR00007	5.5410	1.11913	61
VAR00011	5.5082	1.10488	61
VAR00015	5.0984	1.28718	61

Inter-Item Correlation Matrix

	VAR00003	VAR00007	VAR00011	VAR00015
VAR00003	1.000	.563	.700	.694
VAR00007	.563	1.000	.704	.703
VAR00011	.700	.704	1.000	.831
VAR00015	.694	.703	.831	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	5.332	5.098	5.541	.443	1.087	.051
Item Variances	1.470	1.221	1.750	.530	1.434	.074

Summary Item Statistics

	N of Items
Item Means	4
Item Variances	4

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00003	16.1475	10.261	.717	.533	.896
VAR00007	15.7869	11.404	.719	.542	.891
VAR00011	15.8197	10.717	.851	.741	.848
VAR00015	16.2295	9.680	.843	.737	.846

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
21.3279	18.091	4.25332	4

RELIABILITY

```

/VARIABLES=VAR00004 VAR00008 VAR00012 VAR00016
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	61	100.0
	Excluded ^a	0	.0
	Total	61	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.729	.731	4

Item Statistics

	Mean	Std. Deviation	N
VAR00004	2.7869	1.31801	61
VAR00008	3.3934	1.40569	61
VAR00012	2.6885	1.39691	61
VAR00016	3.7541	1.41015	61

Inter-Item Correlation Matrix

	VAR00004	VAR00008	VAR00012	VAR00016
VAR00004	1.000	.379	.660	.474
VAR00008	.379	1.000	.199	.420
VAR00012	.660	.199	1.000	.299
VAR00016	.474	.420	.299	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	3.156	2.689	3.754	1.066	1.396	.256
Item Variances	1.913	1.737	1.989	.251	1.145	.014

Summary Item Statistics

	N of Items
Item Means	4
Item Variances	4

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item-Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00004	9.8361	9.539	.687	.543	.570
VAR00008	9.2295	11.046	.412	.222	.729
VAR00012	9.9344	10.529	.484	.439	.688
VAR00016	8.8689	10.283	.509	.292	.674

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
12.6230	16.872	4.10757	4

Measurement 1 Combined

RELIABILITY

```

/VARIABLES=VAR00001 VAR00005 VAR00009 VAR00013 VAR00017 VAR00021 VAR00025
VAR00029
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	61	100.0
	Excluded ^a	0	.0
	Total	61	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.938	.939	8

Item Statistics

	Mean	Std. Deviation	N
VAR00001	4.7377	1.50446	61
VAR00005	3.9016	1.49114	61
VAR00009	4.0000	1.75119	61
VAR00013	4.1803	1.42019	61
VAR00017	4.6721	1.43436	61
VAR00021	3.8033	1.38828	61
VAR00025	4.3443	1.54796	61
VAR00029	3.8852	1.52860	61

Inter-Item Correlation Matrix

	VAR00001	VAR00005	VAR00009	VAR00013	VAR00017	VAR00021
VAR00001	1.000	.679	.728	.709	.601	.573
VAR00005	.679	1.000	.804	.693	.538	.659
VAR00009	.728	.804	1.000	.717	.591	.699
VAR00013	.709	.693	.717	1.000	.586	.635
VAR00017	.601	.538	.591	.586	1.000	.729
VAR00021	.573	.659	.699	.635	.729	1.000
VAR00025	.505	.658	.682	.661	.660	.815
VAR00029	.523	.565	.591	.655	.689	.775

Inter-Item Correlation Matrix

	VAR00025	VAR00029
VAR00001	.505	.523
VAR00005	.658	.565
VAR00009	.682	.591
VAR00013	.661	.655
VAR00017	.660	.689
VAR00021	.815	.775
VAR00025	1.000	.728
VAR00029	.728	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	4.191	3.803	4.738	.934	1.246	.131
Item Variances	2.286	1.927	3.067	1.139	1.591	.126

Summary Item Statistics

	N of Items
Item Means	8
Item Variances	8

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00001	28.7869	80.337	.729	.654	.934
VAR00005	29.6230	79.205	.785	.696	.930
VAR00009	29.5246	74.354	.822	.750	.928
VAR00013	29.3443	80.063	.794	.671	.930
VAR00017	28.8525	81.095	.739	.615	.933
VAR00021	29.7213	79.604	.836	.778	.927
VAR00025	29.1803	77.984	.800	.734	.929
VAR00029	29.6393	79.168	.763	.676	.932

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
33.5246	102.254	10.11205	8

RELIABILITY

```

/VARIABLES=VAR00002 VAR00006 VAR00010 VAR00014 VAR00018 VAR00022 VAR00026
VAR00030
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

RELIABILITY

```

/VARIABLES=VAR00001 VAR00005 VAR00009 VAR00013
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	28	100.0
	Excluded ^a	0	.0
	Total	28	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.947	.950	4

Item Statistics

	Mean	Std. Deviation	N
VAR00001	4.7500	1.85841	28
VAR00005	4.0357	1.62121	28
VAR00009	4.0714	1.88421	28
VAR00013	4.1071	1.42307	28

Inter-Item Correlation Matrix

	VAR00001	VAR00005	VAR00009	VAR00013
VAR00001	1.000	.876	.862	.795
VAR00005	.876	1.000	.860	.785
VAR00009	.862	.860	1.000	.784
VAR00013	.795	.785	.784	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	4.241	4.036	4.750	.714	1.177	.116
Item Variances	2.914	2.025	3.550	1.525	1.753	.523

Summary Item Statistics

	N of Items
Item Means	4
Item Variances	4

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item-Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00001	12.2143	21.286	.905	.824	.922
VAR00005	12.9286	23.476	.901	.817	.923
VAR00009	12.8929	21.210	.893	.801	.927
VAR00013	12.8571	26.201	.826	.682	.949

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
16.9643	40.258	6.34491	4

RELIABILITY

```

/VARIABLES=VAR00002 VAR00006 VAR00010 VAR00014
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	28	100.0
	Excluded ^a	0	.0
	Total	28	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.640	.650	4

Item Statistics

	Mean	Std. Deviation	N
VAR00002	4.4643	1.79469	28
VAR00006	4.9643	1.20130	28
VAR00010	2.7857	1.59530	28
VAR00014	4.7143	1.32936	28

Inter-Item Correlation Matrix

	VAR00002	VAR00006	VAR00010	VAR00014
VAR00002	1.000	.266	.230	.617
VAR00006	.266	1.000	.092	.411
VAR00010	.230	.092	1.000	.284
VAR00014	.617	.411	.284	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	4.232	2.786	4.964	2.179	1.782	.972
Item Variances	2.244	1.443	3.221	1.778	2.232	.638

Summary Item Statistics

	N of Items
Item Means	4
Item Variances	4

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00002	12.4643	8.628	.513	.384	.499
VAR00006	11.9643	12.999	.325	.170	.631
VAR00010	14.1429	11.831	.262	.087	.685
VAR00014	12.2143	10.026	.649	.465	.422

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
16.9286	17.254	4.15379	4

RELIABILITY

```

/VARIABLES=VAR00002 VAR00006 VAR00014
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	28	100.0
	Excluded ^a	0	.0
	Total	28	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.685	.694	3

Item Statistics

	Mean	Std. Deviation	N
VAR00002	4.4643	1.79469	28
VAR00006	4.9643	1.20130	28
VAR00014	4.7143	1.32936	28

Inter-Item Correlation Matrix

	VAR00002	VAR00006	VAR00014
VAR00002	1.000	.266	.617
VAR00006	.266	1.000	.411
VAR00014	.617	.411	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	4.714	4.464	4.964	.500	1.112	.063
Item Variances	2.144	1.443	3.221	1.778	2.232	.896

Summary Item Statistics

	N of Items
Item Means	3
Item Variances	3

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00002	9.6786	4.522	.535	.380	.580
VAR00006	9.1786	7.930	.363	.169	.742
VAR00014	9.4286	5.810	.664	.446	.394

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
14.1429	11.831	3.43958	3

RELIABILITY

```

/VARIABLES=VAR00003 VAR00007 VAR00011 VAR00015
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

	N	%
Cases Valid	28	100.0
Excluded ^a	0	.0
Total	28	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.894	.895	4

Item Statistics

	Mean	Std. Deviation	N
VAR00003	5.2857	1.51186	28
VAR00007	5.7857	1.39728	28
VAR00011	5.5000	1.37437	28
VAR00015	4.9286	1.48894	28

Inter-Item Correlation Matrix

	VAR00003	VAR00007	VAR00011	VAR00015
VAR00003	1.000	.679	.838	.651
VAR00007	.679	1.000	.810	.491
VAR00011	.838	.810	1.000	.615
VAR00015	.651	.491	.615	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	5.375	4.929	5.786	.857	1.174	.131
Item Variances	2.086	1.889	2.286	.397	1.210	.038

Summary Item Statistics

	N of Items
Item Means	4
Item Variances	4

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00003	16.2143	13.730	.828	.731	.838
VAR00007	15.7143	15.323	.733	.656	.874
VAR00011	16.0000	14.296	.877	.814	.823
VAR00015	16.5714	15.587	.637	.440	.910

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
21.5000	25.296	5.02954	4

RELIABILITY

/VARIABLES=VAR00004 VAR00008 VAR00012 VAR00016

```

/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	28	100.0
	Excluded ^a	0	.0
	Total	28	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.809	.811	4

Item Statistics

	Mean	Std. Deviation	N
VAR00004	3.5714	1.81411	28
VAR00008	3.9643	1.83550	28
VAR00012	4.1071	1.72861	28
VAR00016	4.1071	1.64067	28

Inter-Item Correlation Matrix

	VAR00004	VAR00008	VAR00012	VAR00016
VAR00004	1.000	.362	.618	.476
VAR00008	.362	1.000	.597	.518
VAR00012	.618	.597	1.000	.531
VAR00016	.476	.518	.531	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	3.938	3.571	4.107	.536	1.150	.064
Item Variances	3.085	2.692	3.369	.677	1.252	.096

Summary Item Statistics

	N of Items
Item Means	4
Item Variances	4

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00004	12.1786	18.967	.577	.416	.784
VAR00008	11.7857	18.693	.587	.416	.780
VAR00012	11.6429	17.720	.733	.554	.708
VAR00016	11.6429	19.720	.615	.381	.766

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
15.7500	31.380	5.60175	4

RELIABILITY

```

/VARIABLES=VAR00004 VAR00008 VAR00012 VAR00016
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Measurement 2

General Course

RELIABILITY

```
/VARIABLES= VAR00001 VAR00005 VAR00009 VAR00013  
/SCALE('ALL VARIABLES') ALL  
/MODEL = ALPHA  
/STATISTICS = DESCRIPTIVE SCALE CORR  
/SUMMARY = TOTAL MEANS VARIANCE.
```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	28	100.0
	Excluded ^a	0	.0
	Total	28	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.947	.950	4

Item Statistics

	Mean	Std. Deviation	N
VAR00001	4.7500	1.85841	28
VAR00005	4.0357	1.62121	28
VAR00009	4.0714	1.88421	28
VAR00013	4.1071	1.42307	28

Inter-Item Correlation Matrix

	VAR00001	VAR00005	VAR00009	VAR00013
VAR00001	1.000	.876	.862	.795
VAR00005	.876	1.000	.860	.785
VAR00009	.862	.860	1.000	.784
VAR00013	.795	.785	.784	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	4.241	4.036	4.750	.714	1.177	.116
Item Variances	2.914	2.025	3.550	1.525	1.753	.523

Summary Item Statistics

	N of Items
Item Means	4
Item Variances	4

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item-Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00001	12.2143	21.286	.905	.824	.922
VAR00005	12.9286	23.476	.901	.817	.923
VAR00009	12.8929	21.210	.893	.801	.927
VAR00013	12.8571	26.201	.826	.682	.949

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
16.9643	40.258	6.34491	4

RELIABILITY

```

/VARIABLES= VAR00002 VAR00006 VAR00010 VAR00014
/SCALE('ALL VARIABLES') ALL
/MODEL = ALPHA
/STATISTICS = DESCRIPTIVE SCALE CORR
/SUMMARY = TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	28	100.0
	Excluded ^a	0	.0
	Total	28	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.640	.650	4

Vs .685

Item Statistics

	Mean	Std. Deviation	N
VAR00002	4.4643	1.79469	28
VAR00006	4.9643	1.20130	28
VAR00010	2.7857	1.59530	28
VAR00014	4.7143	1.32936	28

Inter-Item Correlation Matrix

	VAR00002	VAR00006	VAR00010	VAR00014
VAR00002	1.000	.266	.230	.617
VAR00006	.266	1.000	.092	.411
VAR00010	.230	.092	1.000	.284
VAR00014	.617	.411	.284	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	4.232	2.786	4.964	2.179	1.782	.972
Item Variances	2.244	1.443	3.221	1.778	2.232	.638

Summary Item Statistics

	N of Items
Item Means	4
Item Variances	4

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item-Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00002	12.4643	8.628	.513	.384	.499
VAR00006	11.9643	12.999	.325	.170	.631
VAR00010	14.1429	11.831	.262	.087	.685
VAR00014	12.2143	10.026	.649	.465	.422

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
16.9286	17.254	4.15379	4

RELIABILITY

```

/VARIABLES= VAR00003 VAR00007 VAR00011 VAR00015
/SCALE('ALL VARIABLES') ALL
/MODEL = ALPHA
/STATISTICS = DESCRIPTIVE SCALE CORR
/SUMMARY = TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	28	100.0
	Excluded ^a	0	.0
	Total	28	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.894	.895	4

Item Statistics

	Mean	Std. Deviation	N
VAR00003	5.2857	1.51186	28
VAR00007	5.7857	1.39728	28
VAR00011	5.5000	1.37437	28
VAR00015	4.9286	1.48894	28

Inter-Item Correlation Matrix

	VAR00003	VAR00007	VAR00011	VAR00015
VAR00003	1.000	.679	.838	.651
VAR00007	.679	1.000	.810	.491
VAR00011	.838	.810	1.000	.615
VAR00015	.651	.491	.615	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	5.375	4.929	5.786	.857	1.174	.131
Item Variances	2.086	1.889	2.286	.397	1.210	.038

Summary Item Statistics

	N of Items
Item Means	4
Item Variances	4

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item-Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00003	16.2143	13.730	.828	.731	.838
VAR00007	15.7143	15.323	.733	.656	.874
VAR00011	16.0000	14.296	.877	.814	.823
VAR00015	16.5714	15.587	.637	.440	.910

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
21.5000	25.296	5.02954	4

RELIABILITY

```

/VARIABLES= VAR00004 VAR00008 VAR00012 VAR00016
/SCALE('ALL VARIABLES') ALL
/MODEL = ALPHA
/STATISTICS = DESCRIPTIVE SCALE CORR
/SUMMARY = TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	28	100.0
	Excluded ^a	0	.0
	Total	28	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.809	.811	4

Item Statistics

	Mean	Std. Deviation	N
VAR00004	3.5714	1.81411	28
VAR00008	3.9643	1.83550	28
VAR00012	4.1071	1.72861	28
VAR00016	4.1071	1.64067	28

Inter-Item Correlation Matrix

	VAR00004	VAR00008	VAR00012	VAR00016
VAR00004	1.000	.362	.618	.476
VAR00008	.362	1.000	.597	.518
VAR00012	.618	.597	1.000	.531
VAR00016	.476	.518	.531	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	3.938	3.571	4.107	.536	1.150	.064
Item Variances	3.085	2.692	3.369	.677	1.252	.096

Summary Item Statistics

	N of Items
Item Means	4
Item Variances	4

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item-Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00004	12.1786	18.967	.577	.416	.784
VAR00008	11.7857	18.693	.587	.416	.780
VAR00012	11.6429	17.720	.733	.554	.708
VAR00016	11.6429	19.720	.615	.381	.766

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
15.7500	31.380	5.60175	4

RELIABILITY

```

/VARIABLES= VAR00002 VAR00006 VAR00014
/SCALE('ALL VARIABLES') ALL
/MODEL = ALPHA
/STATISTICS = DESCRIPTIVE SCALE CORR
/SUMMARY = TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	28	100.0
	Excluded ^a	0	.0
	Total	28	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.685	.694	3

Item Statistics

	Mean	Std. Deviation	N
VAR00002	4.4643	1.79469	28
VAR00006	4.9643	1.20130	28
VAR00014	4.7143	1.32936	28

Inter-Item Correlation Matrix

	VAR00002	VAR00006	VAR00014
VAR00002	1.000	.266	.617
VAR00006	.266	1.000	.411
VAR00014	.617	.411	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	4.714	4.464	4.964	.500	1.112	.063
Item Variances	2.144	1.443	3.221	1.778	2.232	.896

Summary Item Statistics

	N of Items
Item Means	3
Item Variances	3

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item-Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00002	9.6786	4.522	.535	.380	.580
VAR00006	9.1786	7.930	.363	.169	.742
VAR00014	9.4286	5.810	.664	.446	.394

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
14.1429	11.831	3.43958	3

Measurement 2 Open Cv

RELIABILITY

```

/VARIABLES=VAR00004 VAR00008 VAR00012 VAR00016
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	28	100.0
	Excluded ^a	0	.0
	Total	28	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.781	.783	4

Item Statistics

	Mean	Std. Deviation	N
VAR00004	4.3214	1.80644	28
VAR00008	4.6429	1.31133	28
VAR00012	3.9643	1.75293	28
VAR00016	4.3929	1.61794	28

Inter-Item Correlation Matrix

	VAR00004	VAR00008	VAR00012	VAR00016
VAR00004	1.000	.488	.752	.437
VAR00008	.488	1.000	.316	.575
VAR00012	.752	.316	1.000	.279
VAR00016	.437	.575	.279	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	4.330	3.964	4.643	.679	1.171	.079
Item Variances	2.668	1.720	3.263	1.544	1.898	.473

Summary Item Statistics

	N of Items
Item Means	4
Item Variances	4

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00004	13.0000	12.889	.742	.648	.638
VAR00008	12.6786	17.856	.560	.402	.748
VAR00012	13.3571	14.905	.577	.570	.735
VAR00016	12.9286	16.587	.499	.364	.772

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
17.3214	25.782	5.07757	4

RELIABILITY

```

/VARIABLES=VAR00003 VAR00007 VAR00011 VAR00015
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	28	100.0
	Excluded ^a	0	.0
	Total	28	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.816	.811	4

Item Statistics

	Mean	Std. Deviation	N
VAR00003	5.6429	1.06160	28
VAR00007	5.7143	1.32936	28
VAR00011	5.8214	1.24881	28
VAR00015	5.5000	1.07152	28

Inter-Item Correlation Matrix

	VAR00003	VAR00007	VAR00011	VAR00015
VAR00003	1.000	.555	.537	.358
VAR00007	.555	1.000	.861	.364
VAR00011	.537	.861	1.000	.429
VAR00015	.358	.364	.429	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	5.670	5.500	5.821	.321	1.058	.018
Item Variances	1.400	1.127	1.767	.640	1.568	.099

Summary Item Statistics

	N of Items
Item Means	4
Item Variances	4

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item-Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00003	17.0357	9.517	.581	.342	.795
VAR00007	16.9643	7.221	.764	.753	.703
VAR00011	16.8571	7.460	.796	.758	.687
VAR00015	17.1786	10.300	.436	.209	.851

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
22.6786	14.448	3.80111	4

RELIABILITY

```

/VARIABLES=VAR00002 VAR00006 VAR00010 VAR00014
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	28	100.0
	Excluded ^a	0	.0
	Total	28	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.735	.732	4

Item Statistics

	Mean	Std. Deviation	N
VAR00002	4.1429	1.45842	28
VAR00006	4.1071	1.47421	28
VAR00010	2.6071	1.22744	28
VAR00014	4.1071	1.28638	28

Inter-Item Correlation Matrix

	VAR00002	VAR00006	VAR00010	VAR00014
VAR00002	1.000	.441	.115	.761
VAR00006	.441	1.000	.454	.423
VAR00010	.115	.454	1.000	.239
VAR00014	.761	.423	.239	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	3.741	2.607	4.143	1.536	1.589	.572
Item Variances	1.865	1.507	2.173	.667	1.442	.112

Summary Item Statistics

	N of Items
Item Means	4
Item Variances	4

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00002	10.8214	9.337	.579	.614	.643
VAR00006	10.8571	9.312	.572	.361	.648
VAR00010	12.3571	12.312	.326	.243	.774
VAR00014	10.8571	9.757	.649	.604	.607

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
14.9643	16.628	4.07778	4

RELIABILITY

```

/VARIABLES=VAR00002 VAR00006 VAR00014
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	28	100.0
	Excluded ^a	0	.0
	Total	28	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.774	.780	3

Item Statistics

	Mean	Std. Deviation	N
VAR00002	4.1429	1.45842	28
VAR00006	4.1071	1.47421	28
VAR00014	4.1071	1.28638	28

Inter-Item Correlation Matrix

	VAR00002	VAR00006	VAR00014
VAR00002	1.000	.441	.761
VAR00006	.441	1.000	.423
VAR00014	.761	.423	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	4.119	4.107	4.143	.036	1.009	.000
Item Variances	1.985	1.655	2.173	.519	1.313	.082

Summary Item Statistics

	N of Items
Item Means	3
Item Variances	3

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00002	8.2143	5.434	.699	.597	.591
VAR00006	8.2500	6.639	.461	.212	.861
VAR00014	8.2500	6.194	.697	.589	.612

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
12.3571	12.312	3.50887	3

RELIABILITY

```

/VARIABLES=VAR00001 VAR00005 VAR00009 VAR00013
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	28	100.0
	Excluded ^a	0	.0
	Total	28	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.759	.771	4

Item Statistics

	Mean	Std. Deviation	N
VAR00001	3.5714	1.66508	28
VAR00005	3.2143	1.19744	28
VAR00009	3.3214	1.33482	28
VAR00013	3.5714	1.39917	28

Inter-Item Correlation Matrix

	VAR00001	VAR00005	VAR00009	VAR00013
VAR00001	1.000	.401	.381	.491
VAR00005	.401	1.000	.859	.278
VAR00009	.381	.859	1.000	.334
VAR00013	.491	.278	.334	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	3.420	3.214	3.571	.357	1.111	.033
Item Variances	1.986	1.434	2.772	1.339	1.934	.322

Summary Item Statistics

	N of Items
Item Means	4
Item Variances	4

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00001	10.1071	10.099	.527	.316	.732
VAR00005	10.4643	11.739	.643	.746	.668
VAR00009	10.3571	10.979	.643	.748	.658
VAR00013	10.1071	12.025	.460	.272	.753

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
13.6786	18.448	4.29516	4

Measurement 2 Combined

Reliability

RELIABILITY

```

/VARIABLES=VAR00001 VAR00005 VAR00009 VAR00013 VAR00017 VAR00021 VAR00025
VAR00029
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	28	100.0
	Excluded ^a	0	.0
	Total	28	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.902	.904	8

Item Statistics

	Mean	Std. Deviation	N
VAR00001	4.7500	1.85841	28
VAR00005	4.0357	1.62121	28
VAR00009	4.0714	1.88421	28
VAR00013	4.1071	1.42307	28
VAR00017	3.5714	1.66508	28
VAR00021	3.2143	1.19744	28
VAR00025	3.3214	1.33482	28
VAR00029	3.5714	1.39917	28

Inter-Item Correlation Matrix

	VAR00001	VAR00005	VAR00009	VAR00013	VAR00017	VAR00021
VAR00001	1.000	.876	.862	.795	.323	.657
VAR00005	.876	1.000	.860	.785	.280	.645
VAR00009	.862	.860	1.000	.784	.352	.732
VAR00013	.795	.785	.784	1.000	.114	.529
VAR00017	.323	.280	.352	.114	1.000	.401
VAR00021	.657	.645	.732	.529	.401	1.000
VAR00025	.735	.696	.815	.742	.381	.859
VAR00029	.199	.203	.307	.136	.491	.278

Inter-Item Correlation Matrix

	VAR00025	VAR00029
VAR00001	.735	.199
VAR00005	.696	.203
VAR00009	.815	.307
VAR00013	.742	.136
VAR00017	.381	.491
VAR00021	.859	.278
VAR00025	1.000	.334
VAR00029	.334	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	3.830	3.214	4.750	1.536	1.478	.256
Item Variances	2.450	1.434	3.550	2.116	2.476	.608

Summary Item Statistics

	N of Items
Item Means	8
Item Variances	8

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00001	25.8929	64.618	.841	.832	.875
VAR00005	26.6071	68.470	.824	.829	.877
VAR00009	26.5714	62.921	.894	.852	.869
VAR00013	26.5357	73.443	.727	.804	.887
VAR00017	27.0714	78.587	.401	.383	.917
VAR00021	27.4286	75.958	.757	.804	.888
VAR00025	27.3214	72.152	.850	.876	.879
VAR00029	27.0714	82.661	.337	.303	.918

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
30.6429	93.201	9.65407	8

RELIABILITY

```

/VARIABLES=VAR00002 VAR00006 VAR00010 VAR00014 VAR00018 VAR00022 VAR00026
VAR00030
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	28	100.0
	Excluded ^a	0	.0
	Total	28	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.797	.800	8

Item Statistics

	Mean	Std. Deviation	N
VAR00002	4.4643	1.79469	28
VAR00006	4.9643	1.20130	28
VAR00010	2.7857	1.59530	28
VAR00014	4.7143	1.32936	28
VAR00018	4.1429	1.45842	28
VAR00022	4.1071	1.47421	28
VAR00026	2.6071	1.22744	28
VAR00030	4.1071	1.28638	28

Inter-Item Correlation Matrix

	VAR00002	VAR00006	VAR00010	VAR00014	VAR00018	VAR00022
VAR00002	1.000	.266	.230	.617	.596	.274
VAR00006	.266	1.000	.092	.411	.278	.190
VAR00010	.230	.092	1.000	.284	.125	.278
VAR00014	.617	.411	.284	1.000	.633	.375
VAR00018	.596	.278	.125	.633	1.000	.441
VAR00022	.274	.190	.278	.375	.441	1.000
VAR00026	.069	.191	.353	.065	.115	.454
VAR00030	.395	.386	.192	.603	.761	.423

Inter-Item Correlation Matrix

	VAR00026	VAR00030
VAR00002	.069	.395
VAR00006	.191	.386
VAR00010	.353	.192
VAR00014	.065	.603
VAR00018	.115	.761
VAR00022	.454	.423
VAR00026	1.000	.239
VAR00030	.239	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	3.987	2.607	4.964	2.357	1.904	.730
Item Variances	2.055	1.443	3.221	1.778	2.232	.362

Summary Item Statistics

	N of Items
Item Means	8
Item Variances	8

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00002	27.4286	38.921	.541	.493	.770
VAR00006	26.9286	46.587	.379	.237	.791
VAR00010	29.1071	44.766	.325	.227	.805
VAR00014	27.1786	40.819	.687	.598	.747
VAR00018	27.7500	39.824	.668	.714	.747
VAR00022	27.7857	42.026	.526	.382	.771
VAR00026	29.2857	47.471	.312	.337	.800
VAR00030	27.7857	41.508	.669	.663	.751

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
31.8929	54.247	7.36528	8

RELIABILITY

```

/VARIABLES=VAR00002 VAR00006 VAR00014 VAR00018 VAR00022 VAR00026 VAR00030
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	28	100.0
	Excluded ^a	0	.0
	Total	28	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.805	.805	7

Item Statistics

	Mean	Std. Deviation	N
VAR00002	4.4643	1.79469	28
VAR00006	4.9643	1.20130	28
VAR00014	4.7143	1.32936	28
VAR00018	4.1429	1.45842	28
VAR00022	4.1071	1.47421	28
VAR00026	2.6071	1.22744	28
VAR00030	4.1071	1.28638	28

Inter-Item Correlation Matrix

	VAR00002	VAR00006	VAR00014	VAR00018	VAR00022	VAR00026
VAR00002	1.000	.266	.617	.596	.274	.069
VAR00006	.266	1.000	.411	.278	.190	.191
VAR00014	.617	.411	1.000	.633	.375	.065
VAR00018	.596	.278	.633	1.000	.441	.115
VAR00022	.274	.190	.375	.441	1.000	.454
VAR00026	.069	.191	.065	.115	.454	1.000
VAR00030	.395	.386	.603	.761	.423	.239

Inter-Item Correlation Matrix

	VAR00030
VAR00002	.395
VAR00006	.386
VAR00014	.603
VAR00018	.761
VAR00022	.423
VAR00026	.239
VAR00030	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	4.158	2.607	4.964	2.357	1.904	.577
Item Variances	1.985	1.443	3.221	1.778	2.232	.377

Summary Item Statistics

	N of Items
Item Means	7
Item Variances	7

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00002	24.6429	30.757	.542	.485	.784
VAR00006	24.1429	37.460	.399	.228	.801
VAR00014	24.3929	32.544	.689	.579	.753
VAR00018	24.9643	30.925	.722	.707	.743
VAR00022	25.0000	33.852	.510	.378	.785
VAR00026	26.5000	39.370	.252	.275	.822
VAR00030	25.0000	32.815	.699	.663	.752

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
29.1071	44.766	6.69073	7

RELIABILITY

```

/VARIABLES=VAR00003 VAR00007 VAR00011 VAR00015 VAR00019 VAR00023 VAR00027
VAR00031
/SCALE('ALL VARIABLES') ALL

```

```

/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	28	100.0
	Excluded ^a	0	.0
	Total	28	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.919	.917	8

Item Statistics

	Mean	Std. Deviation	N
VAR00003	5.2857	1.51186	28
VAR00007	5.7857	1.39728	28
VAR00011	5.5000	1.37437	28
VAR00015	4.9286	1.48894	28
VAR00019	5.6429	1.06160	28
VAR00023	5.7143	1.32936	28
VAR00027	5.8214	1.24881	28
VAR00031	5.5000	1.07152	28

Inter-Item Correlation Matrix

	VAR00003	VAR00007	VAR00011	VAR00015	VAR00019	VAR00023
VAR00003	1.000	.679	.838	.651	.689	.779
VAR00007	.679	1.000	.810	.491	.621	.863
VAR00011	.838	.810	1.000	.615	.635	.811
VAR00015	.651	.491	.615	1.000	.452	.457
VAR00019	.689	.621	.635	.452	1.000	.555
VAR00023	.779	.863	.811	.457	.555	1.000
VAR00027	.675	.805	.744	.451	.537	.861
VAR00031	.343	.346	.352	.070	.358	.364

Inter-Item Correlation Matrix

	VAR00027	VAR00031
VAR00003	.675	.343
VAR00007	.805	.346
VAR00011	.744	.352
VAR00015	.451	.070
VAR00019	.537	.358
VAR00023	.861	.364
VAR00027	1.000	.429
VAR00031	.429	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	5.522	4.929	5.821	.893	1.181	.089
Item Variances	1.743	1.127	2.286	1.159	2.028	.193

Summary Item Statistics

	N of Items
Item Means	8
Item Variances	8

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00003	38.8929	50.618	.850	.831	.898
VAR00007	38.3929	52.321	.837	.828	.899
VAR00011	38.6786	51.782	.886	.817	.895
VAR00015	39.2500	56.343	.565	.517	.924
VAR00019	38.5357	58.925	.683	.564	.913
VAR00023	38.4643	52.851	.857	.879	.898
VAR00027	38.3571	54.608	.814	.775	.902
VAR00031	38.6786	63.634	.375	.260	.932

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
44.1786	71.189	8.43737	8

RELIABILITY

```

/VARIABLES=VAR00004 VAR00008 VAR00012 VAR00016 VAR00020 VAR00024 VAR00028
VAR00032
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA
/STATISTICS=DESCRIPTIVE SCALE CORR
/SUMMARY=TOTAL MEANS VARIANCE.

```

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	28	100.0
	Excluded ^a	0	.0
	Total	28	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.826	.833	8

Item Statistics

	Mean	Std. Deviation	N
VAR00004	3.5714	1.81411	28
VAR00008	3.9643	1.83550	28
VAR00012	4.1071	1.72861	28
VAR00016	4.1071	1.64067	28
VAR00020	4.3214	1.80644	28
VAR00024	4.6429	1.31133	28
VAR00028	3.9643	1.75293	28
VAR00032	4.3929	1.61794	28

Inter-Item Correlation Matrix

	VAR00004	VAR00008	VAR00012	VAR00016	VAR00020	VAR00024
VAR00004	1.000	.362	.618	.476	.089	.338
VAR00008	.362	1.000	.597	.518	.350	.764
VAR00012	.618	.597	1.000	.531	.202	.606
VAR00016	.476	.518	.531	1.000	.125	.414
VAR00020	.089	.350	.202	.125	1.000	.488
VAR00024	.338	.764	.606	.414	.488	1.000
VAR00028	.053	.310	.160	.001	.752	.316
VAR00032	.110	.479	.329	.472	.437	.575

Inter-Item Correlation Matrix

	VAR00028	VAR00032
VAR00004	.053	.110
VAR00008	.310	.479
VAR00012	.160	.329
VAR00016	.001	.472
VAR00020	.752	.437
VAR00024	.316	.575
VAR00028	1.000	.279
VAR00032	.279	1.000

Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance
Item Means	4.134	3.571	4.643	1.071	1.300	.105
Item Variances	2.877	1.720	3.369	1.649	1.959	.293

Summary Item Statistics

	N of Items
Item Means	8
Item Variances	8

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
VAR00004	29.5000	67.444	.413	.445	.826
VAR00008	29.1071	59.507	.712	.663	.782
VAR00012	28.9643	62.554	.640	.597	.793
VAR00016	28.9643	66.332	.524	.507	.809
VAR00020	28.7500	65.231	.498	.666	.813
VAR00024	28.4286	65.291	.756	.720	.786
VAR00028	29.1071	68.840	.382	.609	.829
VAR00032	28.6786	66.004	.548	.477	.806

Scale Statistics

Mean	Variance	Std. Deviation	N of Items
33.0714	83.032	9.11218	8