



UNIVERSITEIT•STELLENBOSCH•UNIVERSITY  
jou kennisvennoot • your knowledge partner

---

21 November 2019

Navrae: Studenteterugvoerkantoor, X3081

Verwysing: 19IW\_THE03

**DOSENT:** Mnr RP Theart  
**DEPARTEMENT:** Elektriese en Elektroniese Ingenieurswese  
**MODULE:** Rekenaarprogrammering 143

Op 11 Oktober 2019 het SOL die terugvoer van 105 studente oor bogenoemde akademiese personeellid en module ontvang. Die terugvoer is op 11 Oktober 2019 afgeneem.

Die terugvoer is ingedeel in sewe kategorieë:

1. Algemeen
2. Biografiese inligting van studente
3. Terugvoer oor die module
4. Geheelindruk van die module
5. Terugvoer oor die dosent
6. Geheelindruk van die dosent
7. Woordelike terugvoer van studente

Die analise van die data is in dieselfde volgorde gedoen. Vir afdelings drie en vyf word terugvoer as 'n gemiddelde punt op 'n kontinuum van een tot vyf gegee. Hierdie resultate word ook grafies voorgestel. Die geheelindruk van die module en dosent word as persentasies uitgedruk. Die ongeredigeerde woordelike terugvoer vanaf studente word aangeheg.

Let asseblief daarop dat die geheelindrukke van die module en dosent, soos aangedui in afdelings 4 en 6, onafhanklik van afdelings 3 en 5 bepaal is.

Vriendelike groete

Veronica Beukes  
Studenteterugvoer  
SOL

Afskrif:

Dekaan:

Departementele Voorsitter:

Ingenieurswese  
Elektriese en Elektroniese  
Ingenieurswese

Module / <i>Module</i> :	Rekenaarprogrammering 143
Departement:	Elektriese en Elektroniese Ingenieurswese
Dosent / <i>Lecturer</i> :	Mnr RP Theart
Datum / <i>Date</i> :	11-10-2019



Aantal respondente: Module <i>Number of respondents: Module</i>	105	Aantal respondente: Dosent <i>Number of respondents: Lecturer</i>	105
--	-----	--	-----

**Biografiese data / Biographical Data**

	Manlik/Male	Vroulik/Female
Geslag <i>Gender</i>	75	30
	71.4%	28.6%

	Afrikaans	Engels/English	isiXhosa	Ander/Other
Moedertaal <i>Mother tongue</i>	16	69	6	14
	15.2%	65.7%	5.7%	13.3%

	50-59%	60 - 69%	70 - 79%	80-89%	90%+
Matriek-gemiddeld <i>Matriculation average</i>	2	2	26	65	10
	1.9%	1.9%	24.8%	61.9%	9.5%

	Selde <i>Rarely</i>	Minder gereeld <i>Less regularly</i>	Altyd <i>Always</i>
Dui aan hoe gereeld jy klas bywoon <i>Indicate how often you attend class</i>	1	20	83
	1.0%	19.2%	79.8%

	1-2 ure / hours	3 - 4 ure / hours	5 - 6 ure / hours	7 - 8 ure / hours	9+ ure / hours
Gemiddelde aantal ure per week buite klas spandeer aan module <i>Average number of hours per week spent outside the class on this module</i>	28	44	24	6	0
	27.5%	43.1%	23.5%	5.9%	0.0%

	Ja/ <i>Yes</i>	Gedeeltelik/ <i>Partially</i>	Nee/ <i>No</i>	Neutraal/ <i>Neutral</i>
Tevrede met die taal van onderrig? <i>Satisfied with the language of tuition?</i>	103	0	1	0
	99%	0%	1.0%	0.0%

	Ja/ <i>Yes</i>	Soms / <i>Sometimes</i>	Nee/ <i>No</i>	Weet nie / <i>Don't know</i>
Is die taal van onderrig in lyn met taalspesifikasies vir die module? <i>Is the language of tuition in the class in line with the language specification for the module?</i>	100	1	1	2
	96.2%	1.0%	1.0%	1.9%

**Module / Module**

	<b>Gemiddeld Average</b> <sup>(1)</sup>	<b>Stadig Slow</b>	<b>Medium</b>	<b>Vinnig Fast</b>	<b>N.v.t. N/A</b>
Tempo van die module: <i>Pace in this module:</i>	3.42	12	45	48	0
		11.4%	42.9%	45.7%	

	<b>Gemiddeld Average</b> <sup>(2)</sup>	<b>Maklik Easy</b>	<b>Medium</b>	<b>Moeilik Difficult</b>	<b>N.v.t. N/A</b>
Moeilikheidsgraad van module, relatief tot ander modules hierdie jaar gevolg: <i>Module difficulty, relative to other modules followed this year, was:</i>	3.2	27	35	43	0
		25.7%	33.3%	41.0%	

	<b>Gemiddeld Average</b> <sup>(3)</sup>	<b>Laag Low</b>	<b>Medium</b>	<b>Hoog High</b>	<b>N.v.t. N/A</b>
Werkklading van module, relatief tot ander modules hierdie jaar gevolg: <i>Module workload, relative to other modules followed this year, was:</i>	3.1	22	55	27	0
		21.2%	52.9%	26.0%	
My vlak van belangstelling in hierdie module, voordat ek daarmee begin het, was: <i>My level of interest in this module, before the start of this module was:</i>	3.6	25	13	66	1
		24.0%	12.5%	63.5%	
My vlak van belangstelling in hierdie module teen die einde van die jaar was: <i>My level of interest in this module, towards the end of the year was:</i>	3.4	24	30	51	0
		22.9%	28.6%	48.6%	

(1) **Gemiddelde tempo gebaseer op 'n skaal van 1 tot 5** (1 = Baie stadig en 5 = Baie vinnig):

Baie stadig en Stadig is saam gegroepeer as **Stadig** en Vinnig en Baie vinnig as **Vinnig**.

**Average pace based on scale 1 to 5** (1 = Very slow and 5 = Very fast)

Very slow and Slow are grouped as **Slow** and Fast and Very Fast as **Fast**.

(2) **Gemiddelde moeilikheidsgraad gebaseer op 'n skaal van 1 tot 5** (1 = Baie maklik en 5 = Baie moeilik):

Baie maklik en Maklik is saam gegroepeer as **Maklik**, en Moeilik en Baie moeilik as **Moeilik**.

**Average difficulty based on scale 1 to 5** (1 = Very easy and 5 = Agree strongly):

Very easy and Easy are grouped as **Easy**, and High and Very High as **High**.

(3) **Gemiddelde gebaseer op 'n skaal van 1 tot 5** (1 = Baie laag en 5 = Baie hoog):

Baie laag en Laag is saam gegroepeer as **Laag** en Hoog en Baie hoog as **Hoog**.

**Average based on scale 1 to 5** (1 = Very Low and 5 = Very High):

Very Low and Low are grouped as **Low** and High and Very High as **High**.

	Gemiddeld Average	Getal / Number			
		Verskil Disagree	Neutraal/ Neutral	Stem saam Agree	N.v.t. N/A
1. Die module uitkomst is aan my gekommunikeer <i>The module outcomes were communicated to me.</i>	4.0	6	19	80	0
2. Die module het gestelde uitkomst bereik. <i>The module achieved its stated aims.</i>	3.8	6	24	75	0
3. Die relevansie van hierdie module tot die program was vir my duidelik. <i>I was able to see the relevance of this module to my programme.</i>	4.1	7	13	85	0
4. Die assessering (bv toetse, werksopdragte) in hierdie module het my gehelp om te leer. <i>Assessment (e.g. test, assignments) in this module assisted me to learn.</i>	3.9	14	12	79	0
5. Ek het voldoende terugvoer oor my werk in hierdie module ontvang. <i>I have received adequate feedback on my work in this module.</i>	3.6	15	27	63	0
6. Duidelike riglyne vir alle assesseringstake is in hierdie module gestel. <i>There were clear guidelines for all assessment tasks in this module.</i>	3.8	12	17	75	0
7. Die bronne vir leer wat vir hierdie module voorsien is (bv die handboek), het my gehelp om effektief te leer. <i>The learning resources provided for this module (e.g. the textbook) helped me to learn effectively.</i>	3.6	15	24	65	0
8. Die keuse van temas en voorbeelde was effektief t.o.v. module-uitkomst. <i>The choice of topics and examples were effective w.r.t. module outcomes.</i>	3.9	8	16	80	0
9. Die fisiese onderrig- en leerruimtes (bv klaskamer groottes, beligting) wat vir hierdie module gebruik is, was voldoende. <i>The teaching and learning spaces (e.g. classroom size, lightning) uses for this module were adequate.</i>	4.0	8	8	88	0
10. Die module was goed gestruktureerd. <i>The module was well structured.</i>	4.0	4	12	88	0
11. Hierdie module het my vaardighede in kritiese denke, analise en oplossing van probleme, kommunikasie ens. help ontwikkel. <i>This module has helped me develop my skills in critical thinking, analysis and problem solving, communication, etc.</i>	4.0	7	21	73	0

(4) Terugvoer oor module op 'n skaal van 1 tot 5 (1= Verskil sterk en 5 = Stem volkome saam).

In die tabel hierbo is Verskil sterk en Verskil saam gegroepeer as **Verskil** en Stem saam en Stem volkome saam as **Stem saam**.

**Feedback on module on a scale of 1 to 5** (1= Disagree strongly and 5 = Agree strongly)

In the table above Disagree strongly and Disagree are grouped as **Disagree**, and Agree and Agree Strongly as **Agree**.

**Geheelindruk van module / General impression of module <sup>(2)</sup>**

<b>Gemiddelde persentasie / Average percentage</b>	<b>72.9</b>		
<b>Verspreiding van module punt</b> <b>Distribution of module mark</b>	Bo gem. Above avg.	Gem. Avg.	Onder gem. Below avg.
Bo gemiddeld >=75%, Gemiddeld 50-74%, Onder gemiddeld <50% Above average >=75%, Average 50 - 74%, Below average <50%	54	35	7

Dosent / Lecturer:

Mnr RP Theart  
Rekenaarprogrammering 143

11-10-2019

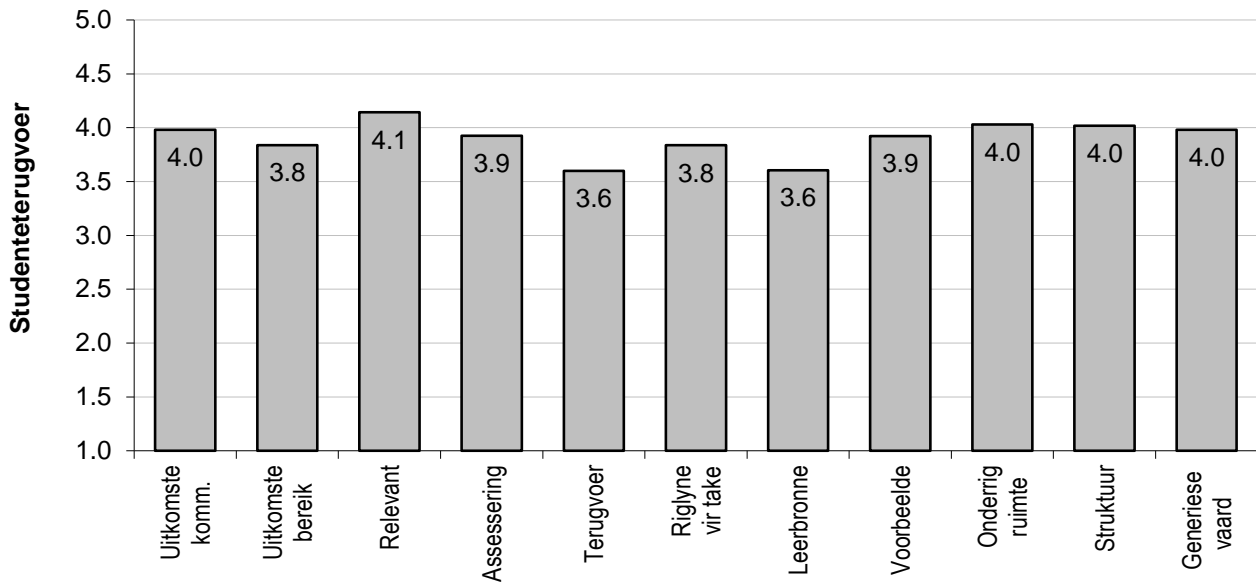
	Gemiddeld Average (5)	Getal / Number			
		Verskil Disagree	Neutraal/ Neutral	Stem saam Agree	N.v.t. N.A.
1. Die dosent was entoesiasies. <i>The lecturer was enthusiastic.</i>	4.2	6	12	87	0
2. Die dosent se verduidelikings was duidelik. <i>The lecturer's explaining was clear.</i>	3.8	7	23	74	0
3. Die lesings was goed gestruktureerd. <i>Lectures were well structured.</i>	4.2	1	12	92	0
4. Die dosent was goed voorberei. <i>The lecturer was well prepared.</i>	4.3	0	11	92	0
5. Terugvoer op take en toetse het stiptelik plaasgevind. <i>Feedback on task and test was given promptly.</i>	3.8	11	23	71	0
6. Studente is aangemoedig om deel te neem aan module (bv vrae vra, voorstelle maak). <i>Students were encouraged to participate in module (e.g. to ask questions, make suggestions)</i>	4.1	1	14	90	0
7. Studente het betekenisvolle terugvoer ontvang. <i>Students received meaningful feedback.</i>	3.8	8	23	73	1
8. Die dosent was toeganklik vir studente. <i>The lecturer was accesible to students.</i>	4.4	2	7	96	0
9. Die dosent was intellektueel stimulerend. <i>The lecturer was intellectually stimulating.</i>	3.9	7	26	71	0
10. Die dosent het my belangstelling in die onderwerp laat verdiep. <i>The lecturer has increased my interest in the subject.</i>	3.6	15	36	53	0
11. Studente in hierdie module is regverdig en met respek behandel. <i>In this module, students were treated fairly and with respect.</i>	4.4	2	7	95	0

(5) Terugvoer oor dosent op 'n skaal van 1 tot 5 (1= Verskil sterk en 5 = Stem volkome saam):

In die tabel hierbo is Verskil sterk en Verskil saam gegroepeer as Verskil en Stem saam en Stem volkome saam as Stem saam.**Feedback on lecturer on a scale of 1 to 5** (1= Disagree strongly and 5 = Agree strongly).In the table above Disagree strongly and Disagree are grouped as **Disagree**, and Agree and Agree Strongly as **Agree**.**Geheelindruk van dosent / General impression of lecturer**

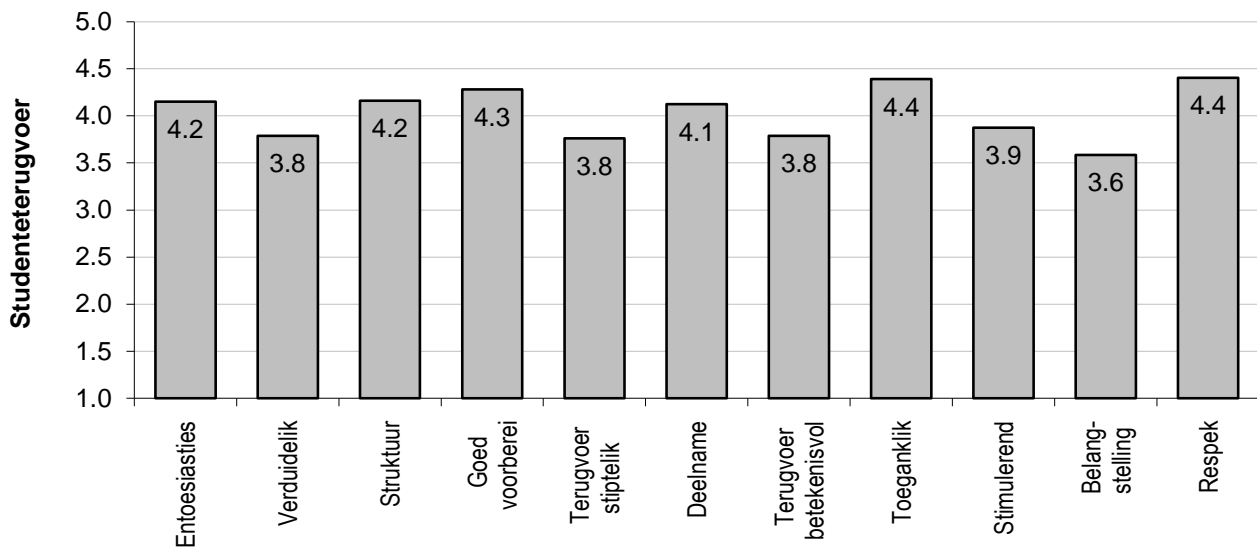
<b>Gemiddelde persentasie / Average percentage</b>	<b>79.8</b>		
<b>Verspreiding van dosente punt Distribution of lecturer mark</b>	Bo gem. Above avg.	Gem. Avg.	Onder gem. Below avg.
Bo gemiddeld >=75%, Gemiddeld 50-74%, Onder gemiddeld <50% Above average >=75%, Average 50 - 74%, Below average <50%	68	21	5

### Gemiddeldes vir Rekenaarprogrammering 143



Aantal respondente = 105

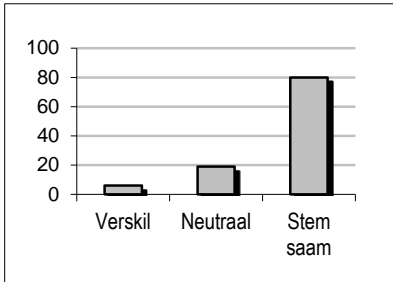
### Gemiddeldes vir Mnr RP Theart



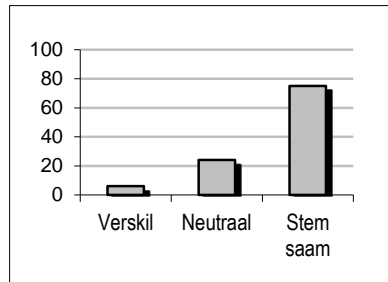
Aantal respondente = 105

## Rekenaarprogrammering 143

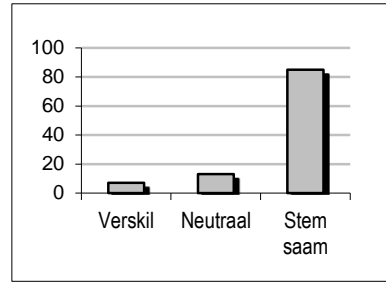
1. Die module uitkomste is aan my gekommunikeer



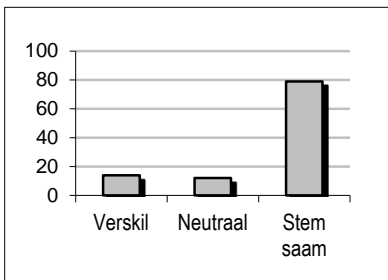
2. Die module het gestelde uitkomste bereik.



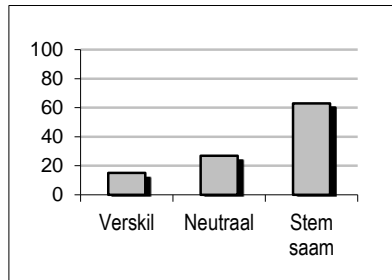
3. Die relevansie van hierdie module tot die program was vir my duidelik.



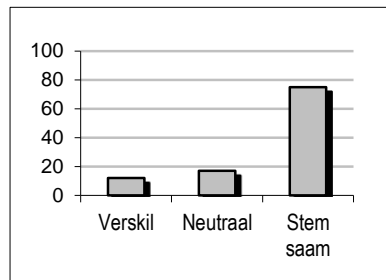
4. Die assessering (bv toetse, werksopdragte) in hierdie module het my gehelp om te leer.



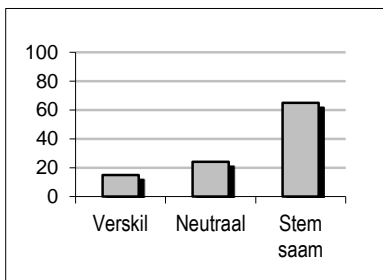
5. Ek het voldoende terugvoer oor my werk in hierdie module ontvang.



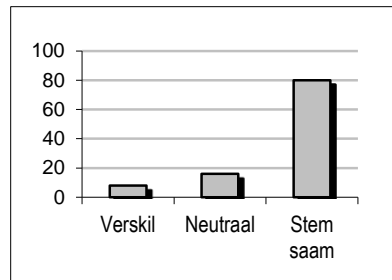
6. Duidelike riglyne vir alle assesseringstake is in hierdie module gestel.



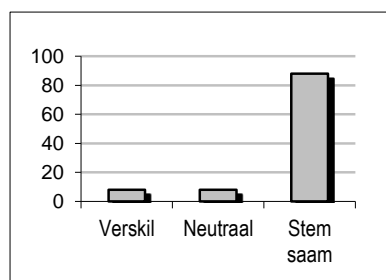
7. Die bronne vir leer wat vir hierdie module voorsien is (bv die handboek), het my gehelp om effektief te leer.



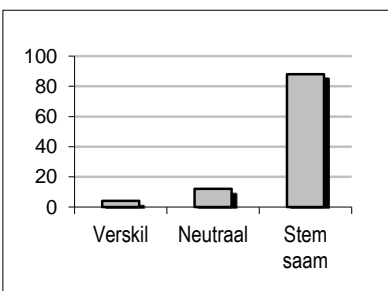
8. Die keuse van temas en voorbeelde was effektief t.o.v. module-uitkomste.



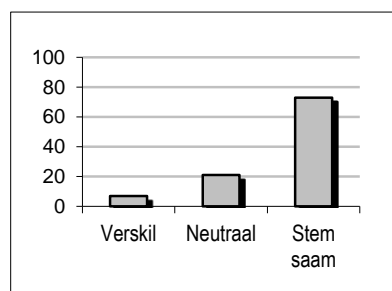
9. Die fisiese onderrig- en leerruimtes (bv klaskamer groottes, beligting) wat vir hierdie module gebruik is, was voldoende.



10. Die module was goed gestruktureerd.



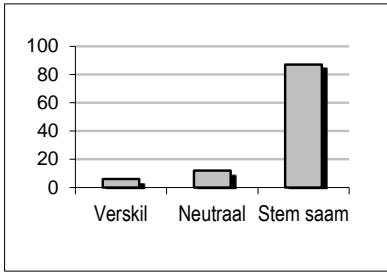
11 Hierdie module het my vaardighede in kritiese denke, analise en oplossing van probleme, kommunikasie ens. help ontwikkel.



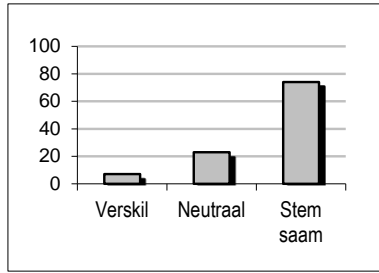
Aantal respondente / Number of respondents: 105

## Mnr RP Theart

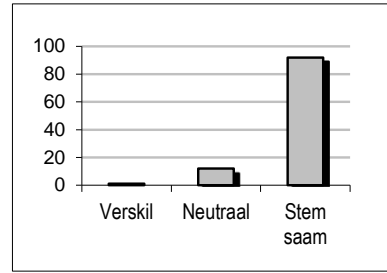
1. Die dosent was entoesiasies.



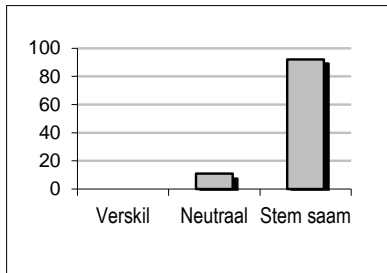
2. Die dosent se verduidelikings was duidelik.



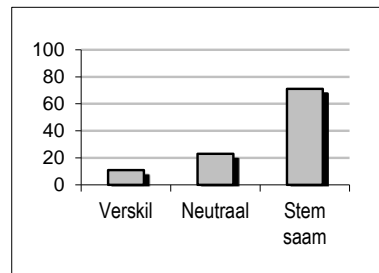
3. Die lesings was goed gestruktureerd.



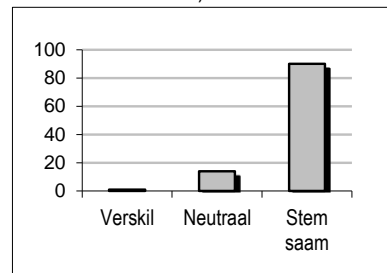
4. Die dosent was goed voorberei.



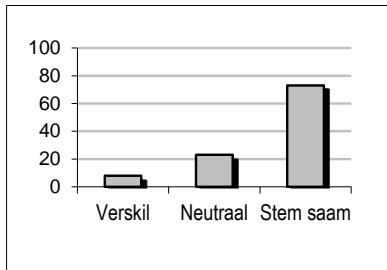
5. Terugvoer op take en toetse het stiptelik plaasgevind.



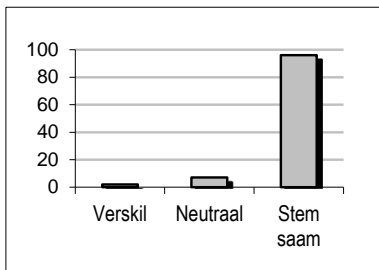
6. Studente is aangemoedig om deel te neem aan module (bv vrae vra, voorstelle maak).



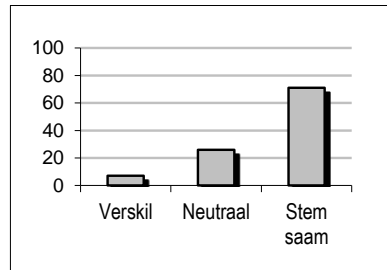
7. Studente het betekenisvolle terugvoer ontvang.



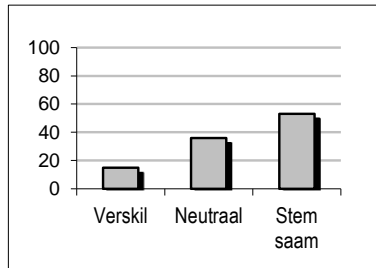
8. Die dosent was toeganklik vir studente.



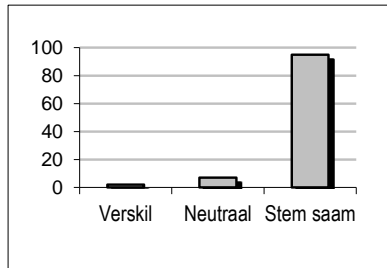
9. Die dosent was intellektueel stimulerend.



10. Die dosent het my belangstelling in die onderwerp laat verdiep.



11. Studente in hierdie module is regverdig en met respek behandel.



**Aantal respondente / Number of respondents: 105**



## **Wat was die beste aspekte van die module Rekenaarprogrammering 143?**

### ***What were the best aspects of the module Rekenaarprogrammering 143?***

---

- It's a practical module.
- Teaching me new interesting stuff.
- Creating programs that could carry out useful tasks.
- Cool programmes that you could create.
- The way problems could be approached from multiple angles.
- Great introduction to programming.
- It's not tedious to code, it's fun.
- New way of thinking.
- Very logical way of thinking.
- Programming.
- Learning new things.
- Learning how a computer works.
- Spare time to work on other modules.
- Learning how to think differently and problem solve.
- Explained in depth.
- The module is interesting, and a very good introduction to its field.
- Interesting, vital in multiple fields.
- I learned something new.
- Learning how to cook.
- Weekly tests are manageable and easy. The practical tutorials gives us good exposure to the type of problem to expect.
- Programming.
- All of them.
- The fact that we got to program.
- Learning a new programming language.
- Pointers.
- Learning how to write programs.
- Learning to make new things.
- Fascinating, good introductory course; appropriate outline and content.
- The problem solving it taught me.
- Just always down and sit down to do some structured logical thinking.
- The pracs.
- The additional slides/notes were great.
- It was very exciting to learn the skills of programming and writing code.
- Learning to code.
- Solving problems.
- Learning about programming and how the computer processes code.
- The practicals.
- Starting to learn how to code and seeing its practical uses.
- When I understood what I was doing.
- Problem solving.
- When I understood and it was fun.
- The fact that it is completely different from other subjects and is a skill that you need to know in today's world.
- The assistance you get during the practical sessions help a lot.
- The module takes into consideration that many have little knowledge about these concepts before.
- Interesting practicals.
- Practical.
- Learning to write/read code.
- Coding my own programs by combining what we learned.
- Learning resources.
- The fact that A1 was a written test, not an actual programming digital test, where you lose, 20% every time.

- The slides are quite well done and computer programming tutorials were worthy.
- Learning to code.
- Practicals help to learn a lot.
- Learning how to programme.
- Developing coding skills.
- The practicals give a clear indication of understanding.
- All of them.
- The enthusiasm from the lecturer.
- Problem solving.
- Practicals.
- I got to learn something very useful and applicable to real life.
- Aided in learning a more widely used programming language as well as developing solving skills which helped in other modules too.
- Practicals.
- Doing practicals.
- The practical parts.
- Already done programming so it was basic (in Matric).
- Learning how to code generally.
- Learning how to develop my skills in programming.
- Improvement of Computer skills.
- Practicals.
- Interesting content.
- Interesting topics.
- Practicals.
- Learning to develop a strong sense of logic, which I find can be applied in everyday life as well as other modules.

## **Watter aspekte van die module Rekenaarprogrammering 143 kan verbeter?**

### ***What aspects of the module Rekenaarprogrammering 143 need improvement?***

---

- Workload.
- The question phrasing in practical tests are too ambiguous sometimes which needs improving.
- Code runner questions should be clearer.
- The tut tests, how questions are phrased and to disregard the minus 20% check.
- Don't use code runner.
- Give a semester project.
- 
- The use of a blackboard, hard to take notes when working on code block cause changes so quickly.
- More time in pracs.
- More accessible examples.
- Demi didn't know the difference between a logical error and syntax.
- More time.
- Time to allow to program at computers to really learn well.
- Way too slow teaching, not enough practice.
- Slides could be improved.
- It needs to be taught on computer and tests should all be programmed on a computer.
- How the questions are set, stating what is needed better.
- When the lecturer does some problems on Code Blocks, they are not on the slides. So there must be a way that the solutions are posted on SUNLearn.
- Should have another day where we could practice programming on computers with assistance of programmers.
- From time to time, the jargon/phrasing used in lectures would get confusing for someone with minimum computer background.
- Linear sorting of arrays.
- No tests in which multiple aspects of programming were integrated into one program.
- How code runner is marked. Usually my code works in codeblocks but it doesn't work in coderunner atleast if I have some correct line of code 4 . Should receive some marks for it.
- For those who have done programming it is nothing new maybe add additional and non-compulsory challenges to the tuts.
- The level of difficulty of the examples. Needs to be simple and gradually increase in difficulty.
- The availability of more difficult examples to practise for people who have never coded before.
- It would be fun to have more exercises (i.e. coding challenges).
- The book delivery system.
- Coderunner.
- Code-runner needs to explained in a better way to reduce submission errors.
- Code runner - should be able to get partial marks.
- Final exams should not be written, they should be in the form of a prac.
- It's very difficult to understand the work when sitting in a lecture taking notes & watching someone else's program. Interactivity with the work is the best way to learn (program with the lecturer).
- How questions in tuts are stated (not clear enough).
- The mic!! (this is for every module).
- The lectures.
- More big problems as the graph in the pointer prac.
- The codes should be uploaded to SUNLearn as slides together with notes.
- We need to have class in a computer room to actually do the things we learn whilst being taught (like CAD).
- It is a bit basic for those who have done programming before.
- Occasional test question discrepancies.
- The way coderunner questions are asked.
- Language we program in, C is good for understanding the principle's but, R Theart's explanations was very good so the medium level language was over compensating. In python.
- Slow down/consolidation is necessary. Need access to see where I went wrong in a test.

- We need more time spent on the computers, you can't teach this subject by using slides and two hours of using the computer.
- The way it is taught, it's hard to learn a 'doing' module, by watching someone do it.
- The module should be taught in a more interactive way (in a computer lab) so students can practise the concepts. Sitting in a classroom for this module is not effective.
- We need to learn in a computer environment so we can practise as we go along.
- Lecture slides and examples posted.
- Everything. Totally new methods needed.
- More relevant examples need to be done in class to prepare us for pracs.
- Helping people who struggle.
- More accessible practical examples.
- The speed at which classes are taught.
- Support outside of class for people who struggle.
- Use of coderunner can be explained more to reduce submission errors.
- 
- The interaction of the lectures with students, the pace to which module is run. Also the slides notes.
- Nothing really, maybe if the lecturers could upload a file to the code they wrote in class it would help as you aren't able to take everything down in time.
- It would be better if we learnt the module in a computer lab so we can practice whilst learning.
- The tut tests and their timing.
- Good for newcomers.
- More indepth. Explaining of the basics.
- Way of teaching.
- The amounts of content covered in one week is a little bit more than what I think it should be.
- The 'coderunner' marking things isn't that great you either get full marks or nothing even if you could have gotten 70%.
- Provide more clear examples.
- Pace was very fast.
- More time for practical tests.
- I think it would perhaps be to the benefit of the student if we were given small logical tests not involving any coding to help develop the skill and understanding required to the module.
- Make the questions in tests (especially prac tests) more clear.

## **Aspekte van dosent se onderrigstyl wat behou moet word.**

### ***Aspects of the lecturer's teaching that should be maintained***

---

- He must teach with enthusiasm.
- The passion for the module content.
- Enthusiasm & knowledge.
- His wider field of knowledge and his ability to demonstrate in real time.
- Enthusiasm.
- Enthusiasm and support especially outside of the classroom.
- Well explained examples in class.
- Eagerness & willingness to help.
- Vivid descriptions/explanations of concepts.
- Accessibility.
- Preparation, enthusiasm.
- Very engaging.
- Enthusiasm and passion.
- Asks class questions and tries to make class fun.
- Enthusiastic teaching style.
- Passion for the module. Respect towards the students.
- His enthusiasm and willingness to help out.
- Enthusiasm as well as general output of work.
- Enthusiasm.
- He should continue working through examples with us using codeblocks.
- Enthusiasm.
- The examples he does and how he explains each part of it.
- Pure explanations.
- Smiling all the time.
- He was enthusiastic and always well prepared and also prepared to admit when he was wrong.
- Enthusiasm and in-depth explanations.
- His use of diagrams to simplify his explanations.
- Use of visual explanations.
- Rensu was a lively and well prepared lecturer, all concepts were well explained, the class atmosphere was calm.
- Showing code for actual programs.
- His easy going nature.
- Explanation of concepts are good.
- Enthusiasm.
- His passion for the module, as well as how determined he is to ensure understanding.
- Actually doing the code on the board, not just explaining it.
- Positive attitude and undoubted passion for programming.
- Enthusiasm for coding.
- His passion for programming.
- Enthusiasm.
- He is very helpful and explains concepts thoroughly.
- Very enthusiastic about the content of the module.
- Interaction with the class and enthusiastic approach to the module.
- Use of examples beneficial to gain a better understanding of the work.
- Approachability.
- Wonderful structure to lessons.
- Level of enthusiasm.
- Enthusiasm of lecturer & his explanations/background of his explanations.
- Enthusiasm for subject.
- Enthusiasm, good slides.
- He is very good at trying to explain, by also writing on the board and trying to run a code multiple times to show us.
- He is very organised and well prepared for the lectures.

- Organised.
- Enthusiasm, encouragement & accessibility to meet with students.
- I think you need to come with a totally new method of teaching.
- Enthusiasm.
- His enthusiasm for the module.
- Enthusiasm. Accessibility to students.
- Enthusiasm and way of explaining.
- Clear explanations, passion for subject.
- Interaction is great.
- His preparation for class.
- Enthusiasm. It's not easy teaching programming speaking from past experiences with IT teachers, but he is very enthusiastic and that helps a lot.
- Worked examples in class.
- Being prepared for a lecture.
- The energy levels.
- Very friendly and approachable.
- Enthusiasm.
- Accessibility and friendliness to students.
- The way he made the class interact with him during the lecture. I felt part of the lecture.
- Everything! Really great at explaining.
- Communication within the classrooms.
- Doing examples in class and allowing students to ask questions.
- Good understanding of content.
- Well structured lectures.
- Explanations and examples and Q&A.
- Enthusiasm and practical application of the modules.
- Making time for students (extra classes, appointments, etc).

## **Aspekte van dosent se onderrigstyl wat verbeter kan word.**

### ***Aspects of the lecturer's teaching that need improvement.***

---

- He must move at a slower pace.
- Explaining some of the concepts.
- Explaining.
- Pace, hard to take notes.
- Less "boring".
- Some find him boring.
- He should involve the class more.
- More detailed work as well as more examples.
- Speed of speech.
- Should talk slowly. Avoid speaking too fast. For someone who is a beginner it's very difficult to follow in class. Reduce talking speed.
- The lecturer was good, but some of the demis were not very friendly and that discouraged me to ask questions at specific times.
- Try make the examples more fun as the boring ones are hard to remember.
- Perhaps more examples code on pointers in class.
- I can't think of anything.
- Needs to be more passionate about lecturing to stimulate the class.
- Slowing down explanations and emphasising important points of explanations.
- 
- Also focussing and explaining the "nitty gritty" stuff, the small things that can cost us marks in the test.
- He talks to the board and not the class.
- Needs to be more enthusiastic.
- Going at a slower pace.
- Sometimes slides are brushed over making it harder to keep up.
- More detailed slides.
- Clarifying certain concepts.
- Perhaps incorporate errors made in tut test into teaching, so we can see what went wrong and how to do it instead.
- Explanation need to be put in a simpler way as lectures explains as if he we're speaking to someone with same knowledge of subject
- Maybe more practical learning/teaching.
- Making the examples more clear and going through them slowly. Not everyone has been exposed to this and therefore fall behind.
- Could try to explain concepts slower and more clearly.
- More interesting/engaging lectures.
- Clearer and more explanations & examples.
- Better examples. Do not give easy examples & give us hard tests. Give hard examples as well.
- Speed - going too fast.
- Should be more direct.
- Attention to the people who struggle.
- Interaction with students, explanations.
- Maybe not refer to the notes so much as we can just do that on our own at home.
- Post examples of the code online.
- Way of teaching.
- Make more examples available on the slide.
- Stop threading like all students know and understand programming. Slow his pace to allow students to sink in the information.
- All good.
- Pace in explaining.
- Focusing perhaps a bit more on the logical side of programming.