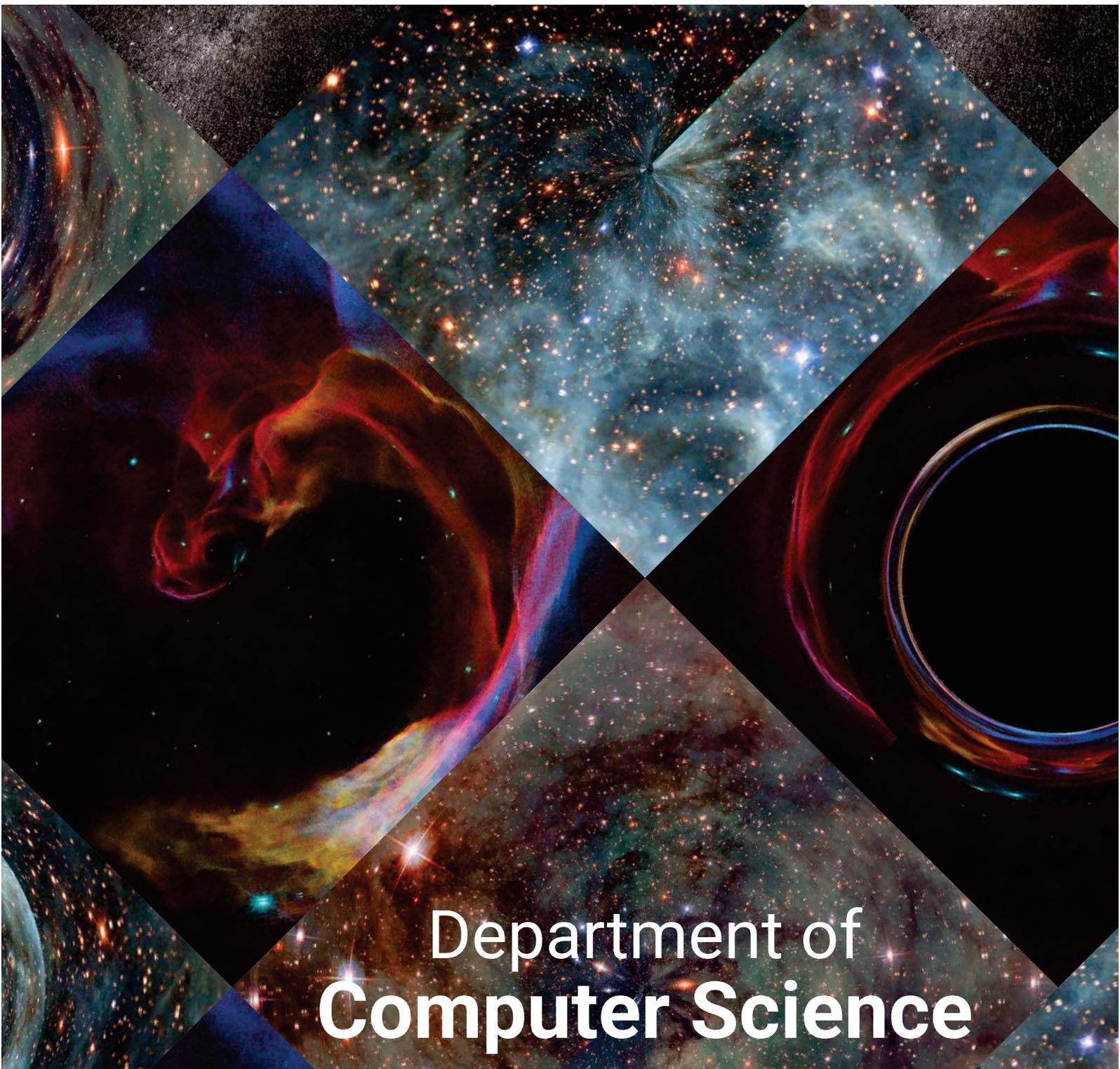




**RHODES UNIVERSITY**

*Where leaders learn*



Department of  
**Computer Science**

student handbook 2026



**RHODES UNIVERSITY**  
*Where leaders learn*

# The Department of Computer Science

Rhodes University has a history of high achievement and is committed to meeting the challenges of the present and future. It is an internationally accepted education centre of excellence, which recognises its southern African setting and the need to meet international standards in an open society. The University is dedicated to cultivating intellects in people who will courageously pursue the truth whatever the circumstances.

Please read this handbook. It is designed to make your life easier  
and facilitate the smooth running of the Department.  
We welcome your comments and suggestions.

Please refer to the online version of this handbook at  
<https://www.ru.ac.za/computerscience/studying/handbook/>  
for updated information about courses and related details.

## Cover Photograph

The research project from which these images were taken focused on implementing a computational approach to black hole optics using raytracing on a GPU. The images show a variety of ray-traced starfield backgrounds with and without black holes. Where no black hole is present, the images present an undistorted grid in flat spacetime; however, the presence of a black hole (with mass  $M = 1.0$ ) in the images shows gravitational lensing and shadow formation. This work was done by Jessica Ackerman.

FIRST YEAR							CS101	
Week	Begins	Mon	Tues	Wed	Thurs	Fri	Topic	
		Orientation week: Monday 2 to Friday 6 Feb						
1	09 Feb						Python Programming, Problem Solving and Computational Thinking interleaved	
2	16 Feb							
3	23 Feb	TEST						
4	2 Mar							
5	9 Mar							
6	16 Mar	TEST						
	23 Mar	Vacation						
7	30 Mar					Good Friday	Python Programming, Problem Solving and Computational Thinking interleaved	
8	6 Apr	Family Day						
9	13 Apr	TEST						
10	20 Apr							
11	27 Apr	Freedom Day				Workers Day		
12	4 May							
13	11 May	TEST						
		Swot Time (16 May - 21 May) & June Exams (22 May - 12 Jun)						
							CS102	
Week	Begins	Mon	Tues	Wed	Thurs	Fri	Topic	
14	6 Jul						Introduction to Java	
15	13 Jul							
16	20 Jul	TEST						
17	27 Jul						Java Programming and Project	
18	3 Aug							
19	10 Aug	Public Holiday	TEST					
	17 Aug	Vacation						
20	24 Aug						Java Programming and Project	
21	31 Aug							
22	7 Sep	TEST						
23	14 Sep							
24	21 Sep				Heritage Day			
25	28 Sep	TEST				Swot week		
		Swot Time (2 Oct - 7 Oct) & October Exams (8 Oct - 5 Nov)						

# First-year mainstream courses

## CS101 & CS102

CS101 is a semester course, which is offered in the first semester of the year. The course is written off at the end of the semester. CS101 is the entry-level course for a major in Computer Science. CS102 is a follow-on course in programming, compulsory for anyone wanting to major in Computer Science in his or her degree. CS102 is offered and examined in the second semester of the year. CS101 and CS102 together make up Computer Science 1.

NOTE: if you wish to continue with Information Systems, you will be required to take CS112 in semester 2, even if you do CS101 and CS102.

### COURSE CO-ORDINATOR

Your course co-ordinator is Mr Marc Marais, Room 105, Hamilton Building (marc.marais@ru.ac.za). All queries and comments about the course should be addressed to your course co-ordinator.

### ENTRANCE REQUIREMENTS

CS101: A rating of at least 5 in Mathematics in the NSC (60%) or an equivalent Mathematics qualification.

CS102: 40% in CS101 in the same year, or having gained credit in CS101 previously.

**NOTE:** Although a 50% pass entitles a student to continue with the next Computer Science course (provided there is space), our experience has shown that students who do not achieve an overall pass of 60% or more in one course are unlikely to succeed in later courses.

It is not necessary to take a course in mathematics as a pre- or co-requisite for CS101, BUT students who wish to major in Computer Science (i.e. take any third year courses in Computer Science) are required to have a credit for at least one semester course in either Mathematics 101 (or Mathematics 1) or Mathematical Statistics 102.

### LECTURES AND PRACTICALS

There are:      5 formal lectures per week  
                    Tests during the semester  
                    1 formal practical session per week (3 hours per session)  
                    Additional after-hours reading and exercises

Lectures:	Monday	1	7:45am
	Tuesday	2	8:40am
	Wednesday	3	9:35am
	Thursday	4	10:30am
	Friday	5	11:25am

Practicals:      Weekday afternoons starting at 14h00, as allocated by the University system.

Practicals will be held in the HAMILTON LABORATORY, provided there is space. You will be allocated to an afternoon practical session and a tutor. You are encouraged to discuss the problems set on practical afternoons with each other, and to build your understanding by seeking help from each other (and, of course, from your tutor). However, you are expected to hand in your OWN INDIVIDUAL SOLUTION, unless you have specifically been asked to work as part of a team. Students who don't abide by this will be dealt with severely. Please read the Plagiarism Policy.

NOTE: It has been shown over many years that students who attend lectures regularly perform better in summative assessments than those with similar abilities. We therefore strongly advise that you attend all timetabled lecture, tutorial, and practical sessions. If you really feel that you can master the course content by teaching yourself, then you should perhaps consider enrolling at a correspondence university, which is likely to be more cost-effective.

### TEXTBOOKS AND HANDOUTS

Course material is available online via RUconnected, and course handouts may be provided.

A non-refundable amount of R250-00 for CS101 and R20 for CS102 will be charged to your University account at the start of the course for class handouts and consumables.

## **COURSE OBJECTIVES AND OUTLINE: CS101**

This course presents basic problem-solving techniques with an emphasis on logical, algorithmic and computational processes using the Python programming language. Students are introduced to programming concepts including output, input, variables, types, operators and operands, debugging, iteration, modules, functions, selection, strings, lists and files.

The prescribed book is “How to Think Like a Computer Scientist: Learning with Python 3” by Peter Wentworth, Jeffrey Elkner, Allen B. Downey, and Chris Meyers, and a printed copy will be provided.

The book also has an interactive edition hosted at <http://runestone.academy>. The interactive edition contains videos, animations, quizzes and even allows you to write code in the book. You will be asked to enrol in a specific course created for CS101 at Runestone Academy. Instructions for doing so will be provided.

## **COURSE OBJECTIVES AND OUTLINE: CS102**

This course further develops programming skills and deals with algorithms, object-oriented principles and practices, and more advanced language features.

The prescribed book is “Think Java: How To Think Like a Computer Scientist” by Chris Mayfield and Allen Downey and a printed copy will be provided. This is a free textbook that can be read online or downloaded for use on your own computer.

## **YOUR TIME COMMITMENT**

The Department of Computer Science expects a CS101 and CS102 student to spend a minimum of **12 hours per week** on Computer Science. This time should be divided approximately as follows:

- 5 hours per week on attending lectures in Computer Science
- 4 hours per week on lecture revision, practical preparation and further reading
- 3 hours per week on practicals

**IMPORTANT:** It is important that you are up-to-date with the material covered in lectures when you arrive at the computer laboratory for your practical session so that you are in a position to get started on the assignment straight away. This is how you will be able to take maximum advantage of the tutorial help available at practical sessions, and ensure that you complete the assignments set for the practical session.

## **ACADEMIC DEVELOPMENT PROGRAMME**

The Academic Development Programme (ADP) exists to enhance student learning in that we provide focused support to students with regard to their weekly practicals and regular course content.

Any student may take advantage of the ADP, but if your lecturers or tutors believe that you have some critical gaps in your learning, you will be required to participate in the ADP. The aim of this programme is to work with you and help you to resolve those minor obstacles that may prevent you from achieving your true potential as a great programmer! If you do not wish to participate in the ADP, you will need to motivate your withdrawal in writing.

To find out more about this programme once the academic year has started, see the course page on RUconnected.

## **DP REGULATIONS**

Please see DP Regulations under Departmental Dynamics.

For your CS101 and CS102 DP to be granted, you are required to maintain an average of at least 40% for your practical tests and achieve an average of at least 40% for your theory tests.

You are expected to collect all your test scripts, make the necessary corrections, and ensure that they are handed to your tutor by the following practical afternoon. Failure to do this may result in you losing your DP.

**WE DON'T TAKE ATTENDANCE INTO CONSIDERATION IF YOUR TEST, PRACTICAL AND EXAM MARKS ARE GOOD. HOWEVER, WE RESERVE THE RIGHT TO TAKE YOUR ATTENDANCE RECORD INTO ACCOUNT FOR BORDERLINE DECISIONS.**

## ASSESSMENT AND SUBMINIMA

- Most practicals are not marked directly, but a practical test is written at the end of the practical to determine the practical mark for the class record.
- The test and examination system is based on a mark per minute.
- The assessment of students in both CS101 and CS102 is based on a total of:
- 20% semester mark (calculated from the results of practicals and test assessments)
- 80% examination (laboratory exercises under examination conditions)
- A course mark of between 40% and 49% allows a student to write a Supplementary examination. Queries in this regard should be directed to your dean.
- If you obtain at least 40% for the course, but have obtained less than 40% for the examination, you will be given a Failed Subminimum (FSM) as your result, and you will not be permitted to aggregate. You will, however, be permitted to write the supplementary examination.
- If you obtain a pass mark for CS1 with at least 40% for CS101 and 50% for CS102 and with no failed subminima, you will be granted a continuing credit (ACR).
- If you obtain a pass mark for CS1 with at least 40% for CS101 and 40% for CS102 and with no failed subminima, you will be granted a non-continuing credit (NCR).

A prize, generously sponsored by  
**Open Box Software,**  
is offered for the top student in the course.

**SECOND YEAR**

**CS201**

Week	Begins	Mon	Tues	Wed	Thurs	Fri	Topic	
		Orientation week: Monday 2 to Friday 6 Feb						
1	09 Feb						Advanced Programming	
2	16 Feb							
3	23 Feb							
4	2 Mar							
5	9 Mar							
6	16 Mar							
	23 Mar	Vacation						
7	30 Mar					Good Friday	Architecture	
8	6 Apr	Family Day						
9	13 Apr							
10	20 Apr						Systems-level Programming	
11	27 Apr	Freedom Day				Workers Day		
12	4 May							
13	11 May							
		Swot Time (16 May - 21 May) & June Exams (22 May - 12 Jun)						

**CS202**

Week	Begins	Mon	Tues	Wed	Thurs	Fri	Topic	
14	6 Jul						Theory of Computation	
15	13 Jul							
16	20 Jul							
17	27 Jul							
18	3 Aug							
19	10 Aug	Public Holiday					Databases	
	17 Aug	Vacation						
20	24 Aug						Databases	
21	31 Aug							
22	7 Sep						Operating Systems	
23	14 Sep							
24	21 Sep				Heritage Day			
25	28 Sep					Swot week		
		Swot Time (2 Oct - 7 Oct) & October Exams (8 Oct - 5 Nov)						

# Second year

## CS201 & CS202

In your first year in the Computer Science Department, you became reasonably confident and competent in designing computer programs. The second-year course builds on these skills and widens your knowledge both of computer systems and programming languages.

At the end of the course, you should be able to code algorithms using the Java and C/C++ programming languages. You should be familiar with advanced and object-oriented programming features, data and file structures, the theory of computation, database methods, operating systems and the background to algorithmic and low-level paradigms.

The CS201 course is offered in the first semester and is examined in June. CS202 is offered in the second semester and examined in November. These two courses may be aggregated to obtain a pass in Computer Science 2 if they are taken in the same calendar year.

### COURSE CO-ORDINATOR

Your CS2 course co-ordinator is **Dr Yusuf Motara**, Room 025, Hamilton Building (y.motara@ru.ac.za). Queries and comments about these courses should be addressed, initially, to your course co-ordinator.

### ENTRANCE REQUIREMENTS

CS201: Minimum of 50% for both CS101 and CS102

CS202: Minimum of 40% in CS201

### LECTURES AND PRACTICALS

There are:           5 Formal lectures per week  
                          1 Formal practical session per week  
                          1 Tutorial session per week, if required  
                          Additional after-hours reading, exercises and essay work.

Lectures:	Monday	3	9:35am
	Tuesday	4	10:30am
	Wednesday	5	11:25am
	Thursday	1	7:45am
	Friday	2	8:40am
Tutorials:	Wednesday	6	12:20am
Practicals:	Wednesday		14:00-17:00

Practicals are held in the UNDERGRAD laboratories in the Hamilton Building on WEDNESDAY afternoons, unless you are notified otherwise. They begin at 14h00 SHARP and officially end at 17h00.

Students will be required to complete practical work amounting to at least 6 hours per week – three hours under formal supervision, and the rest in their own time. You are encouraged to discuss the problems set on practical afternoons with each other, and to seek help from each other (and from the tutors present). However, you are expected to hand in your OWN INDIVIDUAL SOLUTION, unless you have been specifically asked to work as part of a team. Students who copy will be dealt with severely and could lose their DP.

In your second year of Computer Science, you are considered to be a sufficiently experienced programmer to know that you should:

- a) plan your algorithm,
- b) write a well-structured, well-commented program.

NOTE: It has been shown over many years that students who attend lectures regularly perform better in summative assessments than those with similar abilities. We therefore strongly advise that you attend all timetabled lecture, tutorial, and practical sessions. If you really feel that you can master the course content by teaching yourself, then you should perhaps consider enrolling at a correspondence university, which is likely to be more cost-effective.

## TEXTBOOKS, HANDOUTS AND READING LISTS

### CS2 text books

There are NO prescribed textbooks.

Printed or online notes will be provided for all of the modules.

A non-refundable charge of R220-00 for CS201 and R80-00 for CS202 will be made on your university account for printed materials. This includes an amount for paying royalties on copyright material. Several books for additional reading will be referenced during the year, but these will be made available through the library, and you will NOT be required to purchase them. You will be told which books these are and where to find them as you need them.

## COURSE OBJECTIVES AND OUTLINE: CS201

### Advanced Programming

This module builds on the data structures and algorithms covered in Computer Science 1, introducing several new topics. Dynamic data structures are an important focus of the course and underlie the concepts of abstract data types. Important classes of algorithms (such as sorting and searching) are considered and used to introduce simple aspects of algorithm analysis.

### Architecture

This module introduces the logical basis of computer design and how the major components fit together, with emphasis on instruction execution. You will learn how low-level instructions that can be executed directly in hardware relate to code in a higher-level language. The main focus is on understanding the hardware layer by coding in MIPS assembly language. You will also gain an understanding of factors that affect the performance of a computer.

### Systems-level programming

This module introduces the syntax of a high-level language such as C, with emphasis on low-level constructs and on features that have not been encountered thus far. The course builds on knowledge about machine architecture to foster understanding of lower-level, unmanaged languages and how to use them effectively.

## COURSE OBJECTIVES AND OUTLINE: CS202

### Theory of Computation

This module focuses on the more theoretical aspects of Computer Science. It covers computability, complexity, automata, and specifications of formal languages.

### Databases and Software Development Practices

This module introduces core NoSQL concepts, focusing on designing, implementing, and querying MongoDB databases using Python. It provides practical experience through hands-on work in MongoDB, JSON handling, and software development practices in Python.

### Operating Systems

This module covers the fundamental considerations involved in the design and use of a modern operating system. The topics include:

- Introduction to operating systems
- Processes and threads
- Memory management
- Process scheduling
- File management.

## YOUR TIME COMMITMENT

The Department of Computer Science expects a second-year student to spend a minimum of **15 hours per week** on Computer Science 2. This time should be divided approximately as follows:

- 5 hours per week on attending lectures
- 4 hours per week on lecture revision and further reading
- 6 hours per week on practicals

## ASSESSMENT AND SUBMINIMA

### Class Tests

You will be notified at the start of each module which days have been set aside for tests. In addition, spot tests may be set at the discretion of the lecturer. The average test mark is subject to a 40% sub-minimum for DP requirements.

### Practicals and Practical Tests

Correctness of a program is usually worth about 50%, the rest of the marks being awarded for style, readability, ingenuity, documentation, error detection, testing procedures, and example results. Please note that in some modules, the mark for practicals will be obtained from practical tests and not from the practical work submitted. Students must maintain an average of 40% for practical and practical tests in order to obtain a DP.

### Examinations

CS201- Examined in June

Paper 1 (Three hours)	Advanced Programming	85 marks
	Architecture	55 marks
	Systems-level programming	40 marks
		180 marks

CS202- Examined in November

Paper 1 (Three hours)	Theory of Computation	70 marks
	Databases	55 marks
	Operating Systems	55 marks
		180 marks

The assessment of students in CS201 & CS202 is based on a total of:

30% - semester mark, calculated from the results of assignments, practicals, and tests.

70% - allotted to final 3-hour examination.

The examination system is based on a mark per minute in all exams.

### NOTE:

A course mark of between 40% and 49% allows a student to write a Supplementary examination. Queries in this regard should be directed to your Dean.

If you obtain at least 40% for the course, but have obtained less than 40% for the examination, you will be given a Failed Subminimum (FSM) as your result, and you will not be permitted to aggregate. You will, however, be permitted to write the supplementary examination.

A student who has attempted CS201 and CS202 in one calendar year and has failed to pass the assessment for one of the courses, but who has achieved a mark of 50% or better when both courses are added together and the average taken, may be granted an aggregated credit in Computer Science 2. A pass in Computer Science 2 requires that all DP and sub-minimum requirements in both CS201 and CS202 should have been met. Note that the courses may only be aggregated if they are taken in the same calendar year.

A prize, generously sponsored by  
**Janinne Franke,**  
is offered for the top student in the course.

THIRD YEAR							CS301	
Week	Begins	Mon	Tues	Wed	Thurs	Fri	Topic	
		Orientation week: Monday 2 to Friday 6 Feb						
1	09 Feb						Web Technologies and Security	
2	16 Feb							
3	23 Feb							
4	2 Mar							
5	9 Mar							
6	16 Mar						Programming Language Translation	
	23 Mar	Vacation						
7	30 Mar					Good Friday	Programming Language Translation	
8	6 Apr	Family Day						
9	13 Apr							
10	20 Apr							
11	27 Apr	Freedom Day				Workers Day		
12	4 May							
13	11 May							
		Swot Time (16 May - 21 May) & June Exams (22 May - 12 Jun)						
							CS302	
Week	Begins	Mon	Tues	Wed	Thurs	Fri	Topic	
14	6 Jul						Functional Programming	
15	13 Jul							
16	20 Jul							
17	27 Jul							
18	3 Aug							
19	10 Aug	Public Holiday					Machine Learning	
	17 Aug	Vacation						
20	24 Aug						Machine Learning	
21	31 Aug							
22	7 Sep						Networks	
23	14 Sep							
24	21 Sep				Heritage Day			
25	28 Sep					Swot week		
		Swot Time (2 Oct - 7 Oct) & October Exams (8 Oct - 5 Nov)						

# Third year

## CS301 & CS302

In your first year in the Computer Science Department, you became reasonably confident and competent in designing computer programs. CS2 built on these skills and widened your knowledge of data structures, low-level programming, theoretical computer science, and data access using languages such as Java, C/C++ and SQL.

At the end of your third year, you should have a good knowledge of the principles of compilers, networking, practical software development, and web development, and a good introductory knowledge of programming styles and functional programming.

The CS301 course is offered in the first semester and is examined in June. CS302 is offered in the second semester and is examined in November. These two courses may be aggregated to obtain a pass in Computer Science 3 if they are taken in the same calendar year.

### COURSE CO-ORDINATOR

Your CS301 course co-ordinator is **Mr Josh van Staden**, Room 101, Hamilton Building (joshua.vanstaden@ru.ac.za). Queries and comments about a course should be addressed, initially, to your course co-ordinator.

Your CS302 course co-ordinator is **Dr Zelalem Shibeshi**, Room 003, Hamilton Building (z.shibeshi@ru.ac.za). Queries and comments about a course should be addressed, initially, to your course co-ordinator.

### ENTRANCE REQUIREMENTS

CS301 and CS302: Minimum of 40% in both CS201 and CS202, and an aggregate pass in the second year. You do not need to have successfully completed the first semester to continue to the second semester.

Students who wish to major in Computer Science are required to pass a semester course in Mathematics or Statistics.

### LECTURES AND PRACTICALS

There are:           5 Formal lectures per week  
                          1 Formal practical session per week  
                          1 Tutorial session per week, if required  
                          Additional after-hours reading, small projects, tests, exercises and essay work.

Lectures:	Monday 2	8:40am
	Tuesday 3	9:35am
	Wednesday 4	10:30am
	Thursday 5	11:25am
	Friday 1	7:45am
Tutorials:	Thursday 6	12:20am
Practicals:	Thursday	14:00-17:00

All lectures will be held in Room C11 in the GEOLOGY building. Practicals are held in the UNDERGRAD laboratories in the Hamilton Building on THURSDAY afternoons.

NOTE: It has been shown over many years that students who attend lectures regularly perform better in summative assessments than those with similar abilities who do not. We therefore strongly advise that you attend all timetabled lecture, tutorial, and practical sessions. If you really feel that you can master the course content by teaching yourself, then you should perhaps consider enrolling at a correspondence university, which is likely to be more cost-effective.

Students will be required to complete practical work amounting to at least **10 hours** per week (three hours under formal supervision, and the rest in their own time). You are encouraged to discuss the problems with each other, and to seek help from each other (and from the tutors present) when stuck with a problem. However, you are expected to hand in your **OWN INDIVIDUAL SOLUTION**, unless you have been specifically asked to work as part of a team.

**Students who plagiarise will be dealt with severely and could lose their DP.** As third-year students you are reminded to always plan your algorithm and write a well-structured program. In addition, you are encouraged to use one of the document formatting or word processing programs available to produce professional-looking documents for essays, practical write-ups and exercises that have to be handed in for marking.

## TEXTBOOKS, HANDOUTS AND READING LISTS

### CS301 text book

Connolly and Hoar, *Fundamentals of Web Development*, Pearson 2018

### CS302 text book

Kurose, J.F. and Ross, K.W., *Computer Networking: A top-down approach*, 9<sup>th</sup> edition, Pearson, 2025.

Students will also be issued with sets of comprehensive notes. A non-refundable charge of R180-00 for CS301 and R50-00 for CS302 will be made on your university account for such materials. This includes an amount for paying royalties on copyrighted material.

Several books for additional reading will be referenced during the year, but these will be made available on campus and students will NOT be required to purchase them. You will be told which books these are and where to find them as you need them.

## COURSE OBJECTIVES AND OUTLINE

### Web Technologies and Security

This module introduces web technologies used at both the front- and back-end of current Web applications. Front-end technologies that will be discussed include: HTML, CSS and JavaScript. PHP and MySQL (another relational database) will be discussed as examples of technologies for the back end. The security considerations will focus on the core concepts of Web application security primarily focusing on the OWASP top 10 security vulnerabilities and how one can develop a secure website. Topics include the client/server request/response loop, the HTTP protocol, mark-up languages, the Document Object Model, technologies for asynchronous communication and core security considerations. The emphasis of the course, based on the course textbook by Connolly and Hoar, is on the overall architecture of Web applications and how the various technologies help in the construction of applications that go beyond desktop applications.

### Programming Language Translation (PLT)

This module is concerned with the theoretical and practical aspects of programming languages and their translation. A simple stack-based virtual machine (the Parva Virtual Machine) is studied and programmed at the assembler level. This is followed by a discussion of the theory behind modern high-level languages, which leads to the practical aspects of writing compilers, facilitated by the use of compiler writing tools. An in-depth study is made, not only of a compiler for a simple imperative language (Parva, targeting the stack-based PVM), but also of various other syntax-directed software.

The course is based on “Compiling with C# and Java”, by Terry, P.D., published by Pearson, 2005. However, as this book is out of print, a substantial excerpt thereof will be provided as printed course notes.

### Functional Programming

This module introduces a style of programming that avoids state, mutability, inheritance, and nulls and favours purity, immutability, higher-order functions, and options. A strongly-typed cross-platform language will be used for most practical work during this module.

### Machine Learning

This module introduces the fundamentals of machine learning, focusing on data-driven models and the core principles behind them. It relies on key concepts supported by mathematical ideas involving linear algebra and calculus, while integrating applications and hands-on Python work at every stage.

### Networks

This module introduces the fundamental principles of computer networking, primarily focusing on the practical study of the Internet. We will explore areas such as protocol design and the standardisation of computer networks protocols. The module closely follows the structure presented in the course textbook by Kurose & Ross, which guides us through the five traditional layers of the Internet protocol stack, starting from the top, the application layer.

## YOUR TIME COMMITMENT

The Department of Computer Science expects a third-year student to spend a minimum of **20 hours per week** on Computer Science 3. This time should be divided up approximately as follows:

- 5 hours per week on attending lectures
- a minimum of 5 hours per week on lecture revision and further reading
- a minimum of 10 hours per week on practical work

## ASSESSMENT AND SUBMINIMA

### Class Tests

You may be notified about which days have been set aside for tests at the start of each module. In addition, spot tests may be set at the discretion of the lecturer. In some modules, marks are also awarded for short weekly tests on the material of each practical. The average test mark is subject to a 30% sub-minimum for DP requirements.

### Practicals and Practical Tests

Correctness of a program is paramount, though marks may also be awarded or deducted for such things as style, readability, ingenuity, documentation, error detection, testing procedures, and example results. Students must maintain an average of 40% for practical and practical tests to obtain a DP.

### Examinations

CS301 - Examined in June

Paper 1 (Three hours)	Web Technologies and Security	26.9%
Paper 2 (Four hours)	PLT ***	43.1%

CS302 - Examined in November

Paper 1 (Four hours)	Functional Programming	29.2%
Paper 2 (Four hours)	Networks	23.3%
	Machine Learning	17.5%

\*\*\* The CS301 PLT exam has an unconventional format. Students are given part of the paper in advance of the exam to allow them to prepare practical solutions to demonstrate the extent of their knowledge. The venue is a computer laboratory. The content of the whole exam is a mixture of practical and theory. Further details will be available nearer the time.

The assessment of students in CS301 & CS302 is based on a total of:

- 30% semester mark, calculated from the results of assignments, practicals, and tests.
- 70% allotted to final examinations.

### NOTE:

A course mark of between 40% and 49% allows a student to write a Supplementary examination. Queries in this regard should be directed to your Dean.

If you obtain at least 40% for the course, but have obtained less than 40% for the examination, you will be given a Failed Subminimum (FSM) as your result, and you will not be permitted to aggregate. You will however, be permitted to write the supplementary examination.

A student who has attempted CS301 and CS302 in one calendar year and has failed to pass the assessment for one of the courses, but who has achieved a mark of 50% or better when both courses are added together and the average taken, may be granted an aggregated credit in Computer Science 3. A pass in Computer Science 3 requires that all DP and sub-minima requirements in both CS301 and CS302 should have been met. Note that the courses may only be aggregated if they are taken in the same calendar year.

A prize, generously sponsored by  
**BSG Consulting,**  
is offered for the top student in the course.