Tutorial Letter 101/0/2019

Formal Program Verification

COS4892

Year module

School of Computing

This tutorial letter contains important information about your module.

BARCODE

Define tomorrow.

UNISA university of south africa

CONTENTS

Page

Dear Student

1 INTRODUCTION

Welcome to COS4892 in which we study the basic principles of constructing reliable computer programs. This module covers material that is of great importance to the Computer Scientist, Information Technologist, and especially the *Software Engineer*. In the traditional engineering disciplines (Mechanical Engineering, Electronic Engineering etc.) every component of a large system is verified at each stage of the design before it is finally implemented. The aim of this module is to achieve the same precision and reliability for software systems, i.e. develop software that provably achieve a given, mathematical specification.

2 PURPOSE AND OUTCOMES

2.1 Purpose

The construction of provably reliable software requires the use of formal methods based on discrete mathematics and mathematical logic, but it is not very common to find graduates (especially not in South Africa) who possess the appropriate theoretical background. Although the module cannot merely concentrate on providing theoretical background, we attempt in this module to provide the bare minimum of theory, and to treat a relatively large number of examples that illustrate how the theory may be applied. Still, various proof rules for a number of programming constructs are derived and applied to a number of problems. The notions of preconditions and postconditions as part of the specification of a program are central to verification science.

The core of the module is to teach students how to develop the foundation during program design and verification. The logical reasoning will create a base for students to understand how programming can be used and prove to be correct. The abstract nature of the module can be applied on real live problems, which supports the developmental skills of pseudo programming (algorithm development) understanding. The module does not have a specific programming language it focuses on, but teaches mathematical rules and propositions which can be applied in any programming language environment. The main aim of the module is to understand how these propositions work and how to apply them.

2.2 Outcomes

The following are the expected *high-level* outcomes of this module:

Outcome 1

Employ reasoning techniques to solve problems in propositional and first-order logic.

Outcome 2

Apply and discuss various number conversion mechanisms, viz converting real numbers to integers.

Outcome 3

Discharge (i.e. prove) the proof rules developed for the various programming language constructs.

Outcome 4

Apply the various proof rules to a variety of programming constructs to calculate preconditions and to prove the correctness of the said constructs.

Outcome 5

Apply the theory of program verification to develop an algorithm for a medium-sized case study.

Noteworthy is to remember the subject provides students the ability to reason logically, which can be applied when developing algorithms and programming them in a specific programming language. Logical reasoning becomes critical when systems become very large and abstraction levels increase exponentially as is the case in object orientation programming where objects are developed and used in many ways. Understanding logic and structure will support students when they are confronted with these real world problems.

3 LECTURER(S) AND CONTACT DETAILS

3.1 Lecturer(s)

If you have any questions about this module, make contact with the lecturer for this module by:

- either phoning one of the lecturers: +27 11 471 2247 or if unsuccessful, please contact the school at +27 11 670 9200 and ask to speak to the lecturer for the module **COS4892**, or
- by sending an email message to **friedw@unisa.ac.za** or
- by writing to us at the address below:

  The Lecturer (**COS4892**)
  School of Computing
  PO Box 392
  UNISA
  0003

All queries that are not of a purely administrative nature **but** are **about the content of this module** should be directed to us. Please have your study material with you when you contact us.

3.2 Department

This module is offered by the School of Computing. Please note that Computing is a school that has no departments, unlike many other schools in Unisa. The general phone number of the school is +27 11 670 9200 and the school email address is computing@unisa.ac.za.

You should also have access to a computer that is linked to the internet, so you can quickly access resources and information at the University. The *myUnisa* learning management system is Unisa's online campus that will help you to communicate with the university (your lecturers and the administrative departments of Unisa).

To go to the *my*Unisa website, start at the main Unisa website, http://www.unisa.ac.za, and then click on the “Login to *myUnisa*” link on the right-hand side of the screen. This should take you to the *myUnisa* website. You can also access the site directly at: http://my.unisa.ac.za.

3.3 University

If you need to contact the University about matters not related to the content of this module, please consult the publication *Study @ Unisa* that you received with your study material. This brochure contains information on how to contact the University (e.g. to whom you can write for different queries, important telephone and fax numbers, addresses and details of the times certain facilities are open). Also, always include your student number with all written correspondence (email or letter) you may have with the university.

You may first try the following options:

• For students residing in SA, send an sms to	32695
• For all students, send an email request to	info@unisa.ac.za, or
• Fax number (RSA)	012 429 4150
• Fax number (international)	+27 12 429 4150
• E-mail	study-info@unisa.ac.za

4 RESOURCES

4.1 Prescribed book(s)

The prescribed book for this module is:

Backhouse, R. *Program Construction: Calculating Implementations from Specifications*. John Wiley & Sons, 2003, ISBN: 0-470-84882-0.

NOTE: We will refer to the above prescribed book as **RB** or *Backhouse* throughout this tutorial letter and all future material for this module.

You must buy your own copy of the above text book. Prescribed books can be obtained from the University's official booksellers. Please refer to the list of official booksellers and their addresses in the *Study @ Unisa* brochure.

If you have difficulty in locating your prescribed book at these booksellers, please contact the Prescribed Book Section at Tel: 012 429-4152 or e-mail **vospresc@unisa.ac.za**.

4.2 Recommended book(s)

In addition to the above prescribed book, there is also one (1) recommended book for COS4892 which is given below.

Baber, R.L: *The Spine of Software: Designing Provably Correct Software - Theory and Practice*. John Wiley & Sons, Chichester, 1987.

This book may be outdated, but the content is as relevant to computer scientists as it was the day it was published. Baber's book is unfortunately out of print, but the good news is that it is available in its entirety on the web. You can download it free of charge as a PDF file from the following URL:

http://www.cas.mcmaster.ca/~baber/Books/Books.html

The file to look for on the above URL is 'Spine.pdf'. Simply right-click on this file and select '*Save Target As ...*'. Then save this file on your hard disk and print it from there. *Do not attempt to print it directly from the McMaster server, since your PC will most probably stall.*

Important notice

We will not study the whole of Baber's book but only those parts of the theory of program verification not covered by Backhouse.

There are a number of books available from Baber on the above web page.

Additional reading material

The following book may also be consulted for more information or a different perspective. Note that this book may not necessarily be available from the Study Collection of the library and may, therefore, be difficult to obtain:

Gries, D: *The Science of Programming*, Springer-Verlag, 1981.

Recommended books can be requested online, via the Library catalogue.

4.3 Electronic reserves (e-reserves)

There are no e-Reserves for this module.

4.4 Library services and resources

The Unisa Library offers a range of information services and resources:

- for detailed Library information go to http://www.unisa.ac.za/sites/corporate/default/Library
- for research support and services (e.g. personal librarians and literature search services) go to http://www.unisa.ac.za/sites/corporate/default/Library/Library-services/Research-support

The Library has created numerous Library guides: http://libguides.unisa.ac.za

Recommended guides:

- request and download recommended material: http://libguides.unisa.ac.za/request/request
- postgraduate information services: http://libguides.unisa.ac.za/request/postgrad
- finding and using library resources and tools: http://libguides.unisa.ac.za/Research_skills
- Frequently asked questions about the Library: http://libguides.unisa.ac.za/ask
- Services to students living with disabilities: http://libguides.unisa.ac.za/disability

5 STUDENT SUPPORT SERVICES

The *Study @ Unisa* brochure is available on myUnisa: www.unisa.ac.za/brochures/studies

This brochure has all the tips and information you need to succeed at distance learning and, specifically, at Unisa.

There is a student forum for this module on *myUnisa*. This forum is for you to use. I shall monitor the forum from time to time, but as a general rule, I shall refrain from getting involved in your discussions and intervene only when it is really necessary. Nevertheless, please keep the discussions study related.

If you have a question about the content of this module, then please refer to Section 3.1 above.

Additional important information appears in your *Study @ Unisa* brochure.

6 STUDY PLAN

The prescribed book (i.e. *Backhouse* or **RB**) consists of sixteen chapters which must be studied in full. The book is divided roughly into six logical parts:

- Motivation: Chapters 1 and 2 set the scene for why reasoning about algorithms is important.
- Arithmetic preliminaries: Chapters 3, 6, 8, 11 and 12 cover various aspects often needed in the verification of programs. Assignments 01 and 02 cover this part.
- Logic: Chapters 5, 7 and 11 deal with logical reasoning including a thorough treatment of equivalence. Note that Chapter 11 is included in this part as well. Assignments 01 and 02 cover this part.
- Implementation: Chapter 4 is a short chapter and shows how to correctly implement some algorithms in Java.
- Verification principles: Chapters 9, 10 and 13 in **RB** as well as Chapter 3 in **Baber** develop the theory needed to reason about familiar program constructs, e.g. assignments, if-then-else, sequences of statements, loops, etc. Assignment 02 covers this part.
- Practice: Chapters 14, 15 and 16 look at a variety of concrete examples. Assignment 03 covers this part.

Please see *Study @ Unisa* brochure for general time management and planning skills.

7 PRACTICAL WORK

There are no practicals for this module, only the application of theory within assignments.

8 ASSESSMENT

8.1 Assessment plan

COS4892 uses a year-mark system. The mark (percentage) you get for an assignment counts towards your year mark. Every assignment has a total and a weight allocated to it. The weight for Assignment 01 is 0.32 i.e. 32% of your year mark; the weight for Assignment 02 is 0.36 (36%) and the weight for Assignment 03 is 0.32 (32%).

Your year mark (YM) is calculated as: YM = (0.32 * Ass01) + (0.36 * Ass02) + (0.32 * Ass03), where

Ass01 = Percentage obtained for Assignment 01
Ass02 = Percentage obtained for Assignment 02
Ass03 = Percentage obtained for Assignment 03

Your year mark counts 20% towards your final mark and the examination mark makes up the other 80%.

8.2 Assignment numbers

8.2.1 General assignment numbers

Assignments are numbered consecutively per module, starting from 01. The three (3) assignments for this module are number 01, 02 and 03 respectively.

8.2.2 Unique assignment numbers

Assignment number in module	Unique number on system
01	859100
02	649655
03	814216

8.3 Assignment due dates

There are *three* assignments, each of which must be submitted **before or on** its due date. Model solutions to the assignments will be loaded on the Additional Resources for this module after the respective due date.

The due dates appear below:

Assignment no.	Due Date	Chapters
1	31 May 2019	**RB** Chapters 1 - 8
2	26 July 2019	**RB** Chapters 9 - 13 **Baber** Chapter 3

3	27 September 2019	**RB** Chapters 14 - 16

8.4 Submission of assignments

(1) **Compulsory Assignment 1:** Assignment 01 is compulsory for this module and the following apply as far as its submission is concerned:

(a) The due date for Assignment 01 is **31 May 2019**. This assignment **must be submitted before or on this due date in order to count for 32% of your year mark as explained above.**

(b) Submitting the first assignment for this module qualifies you as an "active student" in this module. If you fail to submit Assignment 01 on or before its due date then the university will not allow you to write the COS4892 exam. We cannot enter into any negotiations with you about these regulations. Remember, you are now a responsible postgraduate student who is a good manager of your own time, especially when it comes to submitting your assignments on or before the respective due dates.

(2) Assignments 02 and 03 also count towards your year mark as explained above and must, therefore, be submitted before or on their respective due dates.

PLEASE NOTE: Any assignment received after its due date will be marked in due course but ***will not necessarily earn the full credit***.

(3) If you submit your assignment as a *hard copy*, then, naturally, you need not use the prescribed assignment sheets for your assignments. You are welcome to use a word processor and print your answers provided that you separate the pages, leave a right margin for our comments *and staple all the pages together inside an assignment cover*.

(4) You are encouraged to submit your assignments electronically via the myUnisa website. Please submit in **pdf** format this is the only format allowed by the marking tools that we are using.

(5) Please ensure that you capture the *correct assignment number* for each assignment. Failure to comply with this requirement may cause a considerable delay in the marking of your assignment. Each subsequent page should contain at least the following information:

1. Student number
2. Module code
3. Assignment number

(6) In case you cannot submit your assignment electronically, you may preferably to sending via the post office which should be the last option to consider, go to any unisa's regional office and drop the assignment into any Unisa's assignment box provided for the purpose. Please do not try to submit by fax or e-mail.

(7) Note: It is quite possible that not all assignment questions will be marked. **Please see**

Section 8.6 below for more information.

Please note the following:

For detailed information on assignments, please refer to the *Study @ Unisa* brochure, which you received with your study package.

To submit an assignment via *myUnisa*:

- Go to *myUnisa*.
- Log in with your student number and password.
- Select the module.
- Click on assignments in the menu on the left-hand side of the screen.
- Click on the assignment number you wish to submit.
- Follow the instructions on the screen.

8.5 The assignments

ASSIGNMENT 01

Unique number: 859100
Due Date: 31 May 2019

Chapters to study in **RB**: Chapters 1 – 8

Type: Written/Typed

Total for this assignment: **50**

1. Consider the Knights and Knaves society introduced by RB in Chapter 5. Suppose three people A, B and C are in a conversation and a stranger passed by and asked A: "Are you a knight or a knave?". A answered, but rather indistinctly, so the stranger could not make out the answer. The stranger then asked B: "What did A say?". B replied: "A said he is a knave". Then C says: "Do not believe B; he's lying!".

   Determine what B and C are and what can be deduced about A's answer? Show full reasoning. (8)

2. Consider the following propositions: (12)

   - $(\neg(B \Rightarrow C) \land (\neg(\neg B \Rightarrow (C \lor D)))) \Rightarrow (\neg C \Rightarrow D)$
   - $(A \land \neg(B \lor \neg C)) \Rightarrow (\neg B \Rightarrow (A \land B))$
   - $(\neg A \lor \neg B) \Leftrightarrow (A \Rightarrow \neg B)$

(1) Use the theory introduced in Chapter 5 as well as the equivalences in the Appendix of RB to determine whether the above proposition is a tautology (i.e. always true) or not. Show full workings.

3. Given two propositions p and q, construct the truth table for ($p \equiv q$) and ($p \not\equiv q$). Use your truth table to simplify the following logical expression: (10)

   $p \equiv p \not\equiv q \not\equiv p \equiv q \not\equiv q$

4. Write an essay in which you derive properties for the floor function defined as: (20)

   $N \leq \lfloor x \rfloor \equiv n \leq x$ (**RB** page 73)

Assume that x represents a real number and n is an integer.

Note: your essay should normally start with an **Introduction**, followed by a **Body** and end with a sensible **Conclusion**.

ASSIGNMENT 02

Unique number: 649655
Due Date: 26 July 2019

Chapters to study in **RB**: Chapters 9 – 13

Chapter to study in **Baber**: Chapter 3

Type: Written/Typed

Total for this assignment: **50**

1. Prove the following retro*gressive* theorem for the *if statement* from *Baber* (Q is a precondition while P denotes a post condition): (10)

   If

   {Q1} S1 {P} and {Q2} S2 {P}

   then { (Q1 **and** B) **or** (Q2 **and not** B)} **if** B **then** S1 **else** S2 **endif** {P}

   Use a diagram in your proof.

2. Use the proof rules of Chapters 9 and 10 in **RB** as well as selected portions in Chapter 3 of **Baber** to derive preconditions for the following post conditions and program statements. Show full workings. (12)

   2.1 {?} x:=3-2z {$wy - 2w^2 < z$}

   2.2 {?} x:=x-1; y:=y-1 {$z-1 \leq y < x \leq w$}

2.3 {?}

```
if even(x) → x:= x-1  // x is an even integer
π odd(x) → z:= z+y x  // x is an odd integer
fi
{x≥0 ∧ z+y*x=a*b }
```

2.4 {?} **while** 0 ≥ x ≥ -2 **do** x:=x-1 **endwhile** {x = -3}

Feel free to rewrite any of the above statements into their guarded-command equivalents before calculating their respective preconditions.

3. Explain how to verify a Hoare triple {P} S {Q} with precondition P, program S and post condition Q (3)

4. Verify the following Hoare triple: (5)

4.1 {3 ≤ |x| ≤ 4} **if** x < 0 **then** y:=-x **else** y:=x **endif** {2 ≤ y ≤ 4}

As before, you are welcome to first rewrite the above **if-then-else** statement into its guarded-command equivalent before calculating a new precondition.

5. From the theory developed on pages 184 (bottom) - 186 of **RB** one can infer the following five (5) steps to *prove the correctness of a loop*: (20)

Step 1: Determine the loop invariant (*inv*) by inspection, i.e. by using the material presented in Chapters 12 and 13 of **RB**.

Step 2: Prove that the initialisation establishes the truth of the loop invariant. Refer to Exercise 13.13 in **RB**.

Step 3: Prove that the body of the loop (called *T* by **RB** in equation (13.4) on page 185) preserves the truth of the loop invariant; i.e. prove that the truth of the loop invariant and the truth of the loop condition (i.e. **¬***done*) before execution of the loop body *T* together imply the truth of the loop invariant after execution of *T*.

Step 4: Prove that the truth of the loop invariant and the termination condition which is the negation of the loop condition (i.e. ¬¬*done* ≡ *done*) together imply the correctness of the final result.

Step 5: Prove that the termination condition (i.e. *done*) will be fulfilled in a finite number of executions of the loop body, *T*, i.e. that the loop will terminate in finite time.

Note that the above 5 steps are given on page 87 of **Baber**. The book by Gries (see Section 5.5 above) provides a similar checklist involving the bound function mentioned by **RB**.

Now we are ready to verify the correctness of a loop:

Verify {P} S {Q} where S is the following subprogram which searches an array A for the value of a variable x (Below **and** is the same as ∧):

```
k := 1;
while (k ≤ n and A(k) ≠ x) do k:=k+1 end while
```

The precondition P is (n ∈ ℤ **and** 0 ≤ n) where ℤ is the set of integers and where n is intended to indicate the number of entries in the linear array A.

The post condition Q is:

```
n ∈ ℤ and k ∈ ℤ
and 1 ≤ k ≤ n+1          (this gives the range of k)
and $_{i=1}^{k-1}$ A(i) ≠ x          (so all entries before the k-th ≠ x)
and (k ≤ n and A(k) = x          (i.e. either A(k) = x)
      or k = n+1)          (or no entry in A equals x).
```

In addition the programmer has specified the following loop invariant:

```
n ∈ ℤ and k ∈ ℤ and 1 ≤ k ≤ n+1  (k must lie in its range)
     and $_{i=1}^{k-1}$ A(i) ≠ x          (so we haven't previously found x).
```

Hint: Ignore the bound function *bf* when verifying Steps 1 – 4 above. Use the bound function for verifying Step 5. You may also rewrite the loop as a guarded construct first.

ASSIGNMENT 03

Unique number: 814216
Due Date: 27 September 2019

Chapters to study in **RB**:	Chapters 14 – 16
Type:	Written/Typed
Total for this assignment:	**50**

Write a well-planned essay in which you show how the principles developed in the earlier chapters of **RB** are used to develop an algorithm for Remainder Computation.

Build your essay around the following:

- a formal specification of the problem involving a remainder and quotient
- a development of an elementary version of the algorithm
- extending the development to include a discussion of the **mod** and **div** functions.

Remember that any good essay starts with an **Introduction**, followed by the **Body** of the essay and ends with a sensible **Conclusion**.

8.6 Other assessment methods

Please note the following:

You must answer *all the assignment questions*. **It is possible that not all questions will be marked**. We will make the decision beforehand as to which questions will be marked and students will not be informed of the decision. The same questions will be marked throughout and **only those questions marked will contribute to the percentage you obtain for the particular assignment**.

8.7 The examination

To be admitted to the COS4892 exam you need to submit Assignment 01 on time as explained above. Clearly, since all three assignments count towards your year mark, you should submit assignments 02 and 03 as well, before or on their respective due dates.

Use your *Study @ Unisa* brochure for general examination guidelines and examination preparation guidelines.

8.8 Examination period

This module is offered over a year and the examination is normally during January/February 2019. The Examination Section will provide you with information regarding the examination in general, examination venues, examination dates and examination times at a later stage.

9 FREQUENTLY ASKED QUESTIONS

The *Study @ Unisa* brochure contains an A-Z guide of the most relevant study information.

10 SOURCES CONSULTED

Both the prescribed book and the recommended book were consulted in drafting this tutorial letter.

11 IN CLOSING

Any additional information that may become relevant during the year will be published and announced on *myUnisa*.

I hope you will enjoy this module and find the content useful.

Lecturer:
Dr. Wernher Friedrich

friedw@unisa.ac.za