Computer Science 1015F ~ 2016 ~ Notes to Students

Course Description

Computer Science 1015F and 1016S together constitute a complete Computer Science curriculum for first year students, offering an introduction to the development of algorithms and design of computer programs, along with other selected topics in Computer Science. CSC1015F is offered in the first semester and CSC1016S is offered in the second semester.

Prerequisites

Mathematics at NSC level 6 or better (70%).

No prior knowledge of computer programming is necessary!

Staff

Course Convener:	Dr. Maria Keet	mkeet@cs.uct.ac.za
Lecturers:	A/Prof. Michelle Kuttel	mkuttel@cs.uct.ac.za
	A/Prof. Hussein Suleman	hussein@cs.uct.ac.za
	A/Prof. James Gain	jgain@cs.uct.ac.za
	Dr. Brian DeRenzi	bderenzi@cs.uct.ac.za
Teaching Assistants (TAs):	Yamiko Msosa	yamikom@gmail.com
	Pacome Ambassa	pambassa@cs.uct.ac.za
	Ngoni Choga	Ngoni.Choga@alumni.uct.ac.za
Tutors:	(will be announced on Vula)	

Tentative Schedule of Lectures and Practical Work

The numbers indicate the corresponding chapter of the recommended textbook.

	M	T	W	T	F	Lec	Ptest	Practical	Due Date
15-Feb	Intro	2	2	3	Orient	HS /BdR		Orientation	19-Feb
22-Feb	3	7	7	7				Introduction	26-Feb
29-Feb	7	8	8	8			P. Test 1	1 - I/O, Control (if)	04-Mar
07-Mar	5	5	5	5			P. Test 1	2 - Control (if, while)	11-Mar
14-Mar	*Test 1*	*no lecture*	CS Intro	CS Intro					
21-Mar	holiday	Csintro	6	6	holiday			3 - Control (if, for, while)	25-Mar
28-Mar	Study period								
04-Apr	6	Testing	Testing	Testing		MK/JG	P. Test 2	4 - Fuctions (& strings)	08-Apr
11-Apr	Testing	11	11	11			P. Test 2	5 - Testing	15-Apr
18-Apr	*Test 2 / 11*	13.2	13.2	13.2				6 - Arrays	22-Apr
25-Apr	13.2	13.2	holiday	5.9				7 - Arrays	29-Apr
02-May	holiday	5.9	5.9	13.3			P. Test 3	8 - Recursion	06-May
09-May	*Test 3/ 13.3*	13.1	13.1	Num Sys			P. Test 3	9 - Files	13-May
16-May	Num Sys	Num Sys	Num Sys	Recap					
23-May	Consolidation		Exams						
30-May	Exams								
06-Jun	Exams								
11-Jun	Vacation								

Textbook and Notes

There is *no prescribed textbook*. There is a departmental Python book that is structured roughly in order of the lectures' schedule of topics (see below), available from:

http://www.cs.uct.ac.za/mit_notes/Python/

If you like hard copy books, then the following one is highly recommended:

Python Programming: An Introduction to Computer Science (second edition) *by John Zelle,* Franklin, Beedle and Associates, Inc., ISBN: 9781590282410

Class notes (copies of slides) may be available for selected sections and announced by the relevant lecturer(s). Electronic copies of lecture slides will be made available on Vula.

Vula

Vula (http://vula.uct.ac.za) is the university-wide online learning management system that gives you access to resources to assist in the learning process. The class website for all courses will be located on the Vula system.

Lecturers, TAs and tutors may be consulted through Vula – this is preferable since any questions that are answered may benefit other students as well. Vula is used for the submission of ALL practical assignments and practical tests and for providing students with marks for assignments and tests, and feedback where appropriate. All students will be expected to consult the website on a daily (Monday-Friday) basis for updates on assignments, marks, hints, deadlines, etc.

Refrain from posting anything of the following nature anywhere on the website, as it may violate the university's Appropriate Use of Computer Facilities policy (see ICTS website), necessitating disciplinary and/or legal proceedings: sexist, racist or otherwise discriminatory comments, flame wars, trolling; segments of program code (other than something provided by the instructor); solutions to graded work (or part thereof) before or after submission; or illegal material.

Lectures

For logistical reasons, there are two parallel 'streams' of lectures in the semester. Stream I will be taught by A/Profs Suleman and Kuttel successively, and Stream II will be taught by A/Prof. Gain and Dr. DeRenzi, also successively. They will cover the same material. You are strongly encouraged to stick to the stream you signed up for during registration. Lectures are held in two parallel sessions, being twice in the 4th period and twice in the 5th period.

- Stream I: 4th period (11h00-11h45) in venue NSLT. 5th period (12h00-12h45) in venue CS2A. Monday-Thursday every week and on some Fridays.
- Stream II: 4th period (11h00-11h45) in venue CS2A. 5th period (12h00-12h45) in venue ALEX LT1A (HUM LT1). Monday-Thursday every week and on some Fridays.

We expect the venues for the 4th period to change in the first week of the semester. Please check announcements at least twice daily.

Hotseat

The Computer Science HotSeat is run by senior tutors who can assist you to understand difficult concepts or work through problems you have encountered during lectures, assignments or tests. The HotSeat tutor is available on the ground level opposite the Sci Labs at the other end of the hallway in the CS Building. A schedule of available times will be posted on vula once finalised.

Tests

There will be 3 closed-book 40 minute theory tests. Venues and dates/times will be confirmed in class.

There will be 3 open-book 40 minute practical tests in the Scilabs as part of the laboratory sessions. Each practical test will be offered 2 times – you may write one or both of them and the maximum mark obtained will be used for each test. You have to do the test in the practical session you signed up for.

See the schedule for detailed test dates. These tests contribute to your DP mark and your final mark.

Laboratory Sessions

Laboratory sessions are held in Scilab A/B (Computer Science building, ground floor) and Scilab C (RW James building) on Monday, Tuesday, Wednesday, and Thursday afternoons (2-4pm and 4-6pm) every week, except for March 8 and 9 where the students signed up for Scilab C will have to go to Scilab D (PD Hahn building). You may attend one 2-hour session each week, where you may work on the current practical assignment and discuss any general issues related to practical work with the tutors who are available. Attendance at these sessions is optional. However, in the first hour of a session, students will write a practical test (as described above) if one is scheduled for that week.

It is your responsibility to sign up for a laboratory session that fits in with your timetable. If you do not sign up for a session as soon as possible and find that all slots that suites you best are filled, *it is your responsibility to find a student with whom you can arrange a swop as soon as possible*.

There is an additional 'walk-in' lab session in Scilab A/B on Fridays from 11:00-13:00.

Note, the Scilab C lab sessions of March 8 and March 9 will be in Scilab D (PD Hahn building) instead; please note that this is the PTest1-second-chance week, so try to find it on time. There is no Scilab C session on April 20; those students signed up for that session are allowed to go to another session in that week.

Questions and Submission

All questions for assignments, along with all related files, will be available on the class website on Vula. Practical assignments must be submitted electronically via Vula ONLY. The online submission system used to receive your assignments will provide the official timestamp used to determine whether a program is on time. Marks will be deducted automatically for automatically-marked assignments that are submitted late.

Marking

Most assignments will be marked automatically based on test cases and the marks will be uploaded to Vula. Tutors will mark randomly-chosen practical assignments during the semester.

Equipment and Programming Language

All programming will be done in Python v3 unless otherwise stated.

It is the responsibility of the student to submit a program that will successfully execute on the specified platform. Any student who works on their own equipment must ensure that all assignments will execute on the university equipment before submission – no discussion will be entered into after submission.

Computing facilities are available for use in the Scilabs that are located in the Computer Science building (Scilab A and B), RW James (Scilab C) and P D Hahn (Scilab D). Students also may use The Shuttleworth Lab, which is located in the Computer Science building and is open 24/7 (with student-card access).

It is ALWAYS the student's responsibility to ensure that adequate backup copies are made of all work in progress and all work already completed. Loss of data or programs is not an acceptable excuse for non-submission or late submission of assignments.

Plagiarism

Refer to attached document for the departmental plagiarism policy. *This policy will be strictly enforced*.

All assignments, tests and examinations are done individually – there is NO group work allowed in this course. It is acceptable to discuss the questions for assignments with peers but not the specific details of the solutions, nor engage in "pair programming". When in doubt, speak to a tutor or TA.

Students are required to sign and submit a form (on the last page) verifying that they have read and understood the contents of this policy before commencing any form of assessed work.

There will be randomised code plagiarism checking.

DP Requirement

A student is granted DP status (and therewith may write the exam) in CSC1015F if the following condition is met:

- (3/5 * Assignment average + 2/5 * Practical test average) >= 45%

Final Examination

The examination timetable will be published sufficiently in advance of the final examination on university notice boards. It is the student's responsibility to take note of the correct time and place for the examinations.

All examinations will be cumulative, closed-book and closed-notes (i.e., you may not bring your notes or textbooks into the examination room), and 2 hours in duration.

A final mark in CSC1015F will be calculated as follows:

Final = 0.15 * Assignment average + 0.15 * Test average+ 0.10 * Practical test average + 0.60 * Exam

In order to pass, ALL of the following requirements MUST be met:

- Final >= 50%

- (3/5 * Practical average + 2/5 * Practical test average) >= 45%
- (1/5 * Test average + 4/5 * Exam) >= 45%

All students who pass CSC1015F are eligible to continue with CSC1016S, as are those students who are granted a supplementary exam.

Supplementary Examinations

Students who do not pass but obtain a mark of 45-49 may be awarded a supplementary examination; this is indicated either with *x*S (with 45 <= *x* <= 49) or OSS ['failed theory subminimum'] in PeopleSoft. These are written in January and the exams office will contact those eligible for it, including the exact time and date of the supplementary exam.

The final mark will be calculated as above, but then with the supplementary exam mark. The system then converts this into a "UP" unqualified pass or "UF" unqualified fail if you had a plain supp and into the 'June grade' if you had an OSS with the June exam.

Please also note the following (short version of some of the handbook rules): there is no supp on a supp exam, no deferring a supp exam, no supp on a deferred exam, and no deferring a deferred exam.

Grade Allocation

1 = 75-100; 2+ = 70-74; 2- = 60-69; 3 = 50-59; F = 0-49.

Information Dissemination and Communication

Attendance and Absence

This is a lecture course. While attendance at lectures is not mandatory after the first day, all marked work (assignments, tests, and exams) will be based on the lectures. Obviously, non-attendance at tests and exams will result in a mark of 0 (zero).

ALL students will be expected to complete ALL assigned work. If you miss ANY assigned work with a legitimate reason, send an email to the course convener within a week or as soon as possible thereafter. Note that there are few legitimate reasons that will be accepted – these include hospitalization or illness – and a medical certificate from a qualified medical practitioner is typically required. Such medical certificates must be delivered to the departmental secretary.

Queries

Any queries about the *content* of the lectures MUST be directed to the lecturer teaching that section.

Any queries about *marks* or *marking* of practical assignments must be directed to your tutor or TA.

All marked work (whether in paper or electronic format) must be kept until the end of the semester. In general, queries about marks MUST be made within a week of marked work being returned. No queries about any marks will be entertained after the final examination.

Any queries about the *administration* of the course must be directed to the TA.

The course convener must ONLY be contacted as a last resort, unless otherwise indicated.

Disability

If any student needs special accommodation because of a disability, please contact the course convener during the first week of classes.

Syllabus

Corresponding chapter numbers in the recommended textbook are indicated in parentheses.

- Introduction to Computer Science (Ch. 1): What is Computer Science, Applications of Computing, History of Computing, Computer Hardware (Machine Architecture), Computer Software (System Software, Applications), Algorithms, Programming Languages
- Introduction to Python Syntax (Ch. 2.1-2.5; 3): Basic syntax, variables, operators, comments, expressions, output
- Conditionals (Ch. 7): Boolean expressions and logical conditions, If statements, nested ifs, if-else, if ladders,
- Loops (Ch. 8): for, while, nested loops
- Strings and Input (Ch. 5)
- Functions (Ch. 6): parameters, return values

- Testing: debugging, equivalence classes
- Arrays (Ch. 5; 11): lists, dictionaries, sets, multi-dimensional arrays
- Recursion (Ch. 13.2)
- Sorting and Searching (Ch. 13.1; 13.3)
- File I/O (Ch. 5.9): text files, exceptions
- Number Systems: Machine representations of data, Binary operations, Boolean algebra

Science Extended Degree Programme

The Computer Science Department runs academic development courses – CSC1010H and CSC1011H – that are part of the Extended Degree Programme in Science (EDP). CSC1010H runs over approximately an entire year and covers the syllabus of CSC1015F at a somewhat slower pace and in greater depth. In the first semester of the following year of study, CSC1011H covers additional material to assist students in their transition to CSC1016S, which forms the second semester module of CSC1011H. Note that each of these EDP courses counts only as a half credit.

Students who do not perform adequately in the first test in CSC1015F will be invited to join CSC1010H at the beginning of the second quarter, on condition that the student has a good record of submitting practical assignments and there is sufficient space in the CSC1010H course. Any student who feels that they may benefit by joining CSC1010H should consult Maria Keet (Course convener) or Gary Stewart (EDP Coordinator) before the end of the first quarter.

Extra Problems: the CSC1015 Challenge

This course is about problem solving, but for some of you, the exercises and assignment problems may be not enough, be this because you already may know how to program or you had rather seen more real-life problems that can be solved using a computer. Therefore, we have decided to add a challenge. This is not for marks, but you can gain (small) prizes with it. It is also good practice for the IT Challenge team programming contest. You should *only* attempt this *after* you've *successfully* completed *all* your homework and assignments and tests. Note also that tutors will *not* assist you in solving the problem.

The setting and rules are as follows.

- There are 12 problems to solve, which are of varying level of difficulty. The points you can score is an indication of difficulty and/or amount of code needed for the solution (more points = harder or more work). The topics of the problems range form a few problems for fun (lazily unpacking your belongings in res) to realistic ones (e.g., smart homes with user-mediated electricity 'load balancing', calculating the level of social unrest).
- The programming techniques you will need to implement the solution are to a greater or lesser extent covered in the course (arrays, if-else etc.). The skills and techniques needed to solve the problems match the order of the course's syllabus for at least several problems, but not all.
- As with assignments and prac tests, the automatic marker will be used. However, the only information in the report is "wrong answer" or "accepted"; that is, unlike with the assignments, you do not get to see the test cases. The marking is an all-or-nothing: either all test cases complete successfully and you will get the points, or not and have one penalty point deducted.
- The system only accepts solutions written in Python.
- It is not a weekly race against the clock, but a final deadline there is: you have until May 17 to submit solutions. The release dates of successive sets of questions are: March 7, March 21, April 11, and April 25.
- There will be essentially 3 scoreboards, merged into one in the leaderboard interface: 1) general leaderboard based on points accumulated having solved one or more problems, 2) a first-to-solve-the-problem scoreboard (colour-coded in green), and 3) a first *new* student to solve a problem scoreboard (colour-coded maroon). On the latter, e.g., Joe Soap is the first to solve Problems A, B, and C, in that order, then that will appear as such in the scoreboard in green. Then Joanne Soap solves Problem C, which causes the scoreboard to update with a maroon-solved. Other solved problems will have a blue colour. Any other solved problem is colour-coded with blue. When a potential solution is submitted but was not fully right yet, then it will turn up as a pink box.
- To keep some suspense, the scoreboard updates occur only once a day.
- The tutors will not help you solve these problems.
- Besides the honour to be on the scoreboards, there is a small prize for each first new student to solve a problem, and less small prizes at the end for the top-5 of the general leaderboard.

Finally, this is experimental. The Challenge will run on a separate vula site.

Department of Computer Science
University of Cape Town

Policy on Academic Dishonesty for Computer Program Submissions

Introduction

The University of Cape Town has well-defined policies on copying and plagiarism, which are contained in the general Rules for Students and set out in full on the UCT website at: http://www.uct.ac.za/about/policies/. The Department of Computer Science has set out the following guide and interpretation of these rules and polices as they apply to courses involving computer programming, and the use of the computer. The rules defined here are in line with the best procedures in other institutions and, in particular, are adapted from those on academic dishonesty at Oregon State University[1] and academic dishonesty and cheating at San Francisco State University[2].

The purpose of teaching and learning in the Computer Science Department is to enable you, the student, to understand the work that you have submitted. Thus, you are required to do your own work for an assignment in order to ensure that you understand it.

Academic Dishonesty for Computer Program Submissions

Academic dishonesty is EITHER:

- Copying and plagiarism of computer program submissions, where you present, as your own work, material produced by or in collaboration with others, without proper acknowledgment, and where implicitly or explicitly disallowed for the assigned work.

OR

- Permitting or assisting others to present your work as their own.

Guidelines for Determining Academic Dishonesty

The following guidelines are provided to help in determining if an incident of academic dishonesty has occurred. A member of staff may suspect a student of academic dishonesty if the student submits a program that is so similar to the program submitted by a present or past student, in whole or in part, in the course that the solutions may be converted to one another by a simple mechanical transformation. A member of staff may suspect a student of academic dishonesty, whether on a program or an examination, if the student cannot explain both the intricacies of his or her solution and the techniques and principles used to generate that solution.

In all circumstances, it is acceptable to discuss the meaning of assignments and general approaches and strategies for handling those assignments with other members of the Academic Community.

Any cooperation beyond that point, including shared pseudocode or flowcharts, shared code, or shared documentation, is only acceptable if specifically so permitted by the lecturer in written guidelines distributed to the entire class, either through printed handouts or on the course website or other official communications channel.

[1] http://cs.oregonstate.edu/acad/policies/dishonesty.htm

[2] http://cs.sfsu.edu/plagarism.html

Where a student working on an individual assignment includes the work of others (e.g., algorithms transcribed from a textbook), the student must clearly indicate in the code what part(s) are the work of others.

Penalty for Academic Dishonesty

In all cases where academic dishonesty is suspected, is tested, and is deemed to have occurred, no credit will be given for the assignment in question and a record will be kept in the Department of Computer Science. A report may also be submitted to the Vice-Chancellor's office for possible disciplinary action through the University Disciplinary Tribunal, which may include expulsion, suspension, or probation, as well as lesser sanctions. In all cases, students will have the right to be heard by the lecturer, the course convener and the Head of Department before any action is finalized.

Identifying Copying and Plagiarism in Computer Program Submissions

As the Department of Computer Science believes strongly that students should only get credit for their own work, it will take all necessary steps to identify any cases of copying and plagiarism of computer submissions. These could include:

- Submitting program files to plagiarism detection sites, either locally or internationally.
- Setting a test on the techniques used in a program.
- Requesting students to explain sections of their code.

If you have any doubt or if you have any questions, you should consult with your lecturer or course convenor as to whether or not your work with other students and programs prepared for submission are appropriate.

Examples of Academic Dishonesty

The following examples illustrate situations when Academic Dishonesty has and has not occurred. Please note that these lists are not comprehensive!

Academic Dishonesty has occurred:

- When a student turns in the work of another student and represents it as his or her own work.
- When a student includes another person's work in their own submission without setting out in full what part(s) are the work of others.
- When a student knowingly permits another to turn in his or her work, including by providing access to personal files through sharing of login credentials.
- When a student copies code from the work of another student.
- When a student deliberately transforms borrowed sections of code in order to disguise their origin.
- When several students collaborate on a project (whether or not group work is allowed) and fail to inform the lecturer of this.
- When a student steals, obtains solutions, or program samples from another student's output or personal files.
- When a student is unable to explain the working of a piece of code.

Academic Dishonesty has NOT occurred:

- When students have permission to collaborate on a project, and list all collaborators.

- When students receive advice from tutors, teaching assistants, or staff members involved in the course.
- When students share knowledge about syntax errors, coding tricks, or other language-specific information that makes programming easier, except where such techniques represent the core topics being tested in the assignment.
- When students engage in a general discussion of the nature of an assignment, the requirements for an assignment, or general implementation strategies.
- When students engage in discussion of course concepts or programming strategies in preparation for an assignment or examination.
- When students copy code and cite its source on assignments for which the lecturer allows inclusion of code other than the student's own.

Conclusion

The Computer Science Department believes that encouraging students to submit their own work and actively enforcing this policy will ensure that work done is appropriately rewarded. All students will benefit from the ensuing recognition of the high standard of graduates from our programme.

Document History

v1, 2003:

First version of policy

v2, February 2016:

Revision of policy to reflect practice and changes in operations

Computer Science 1015F

2016

Plagiarism Policy Acceptance

I, ____, Student number: _________________________, hereby acknowledge that I have read and understood the plagiarism policy of the Department of Computer Science. I will adhere to this policy and the general policies of the university referred to therein.

Signature: ____

Date: __________________________________