

Introduction to Computer Program Design - Course Outline

COMP 102: 2016 Trimester 1

This document sets out the workload and assessment requirements for COMP 102. It also provides contact information for staff involved in the course. If the contents of this document are altered during the course, you will be advised of the change by an announcement in lectures and/or on the course web site. A printed copy of this document is held in the School Office.

COMP 102 is about the design and construction of computer programs. The course forms a basis for later courses in Computer Science and Engineering, but will also be useful for students who want to learn computer programming but do not intend to study this subject at higher levels.

Prescription

This course introduces the fundamentals of programming in a high-level programming language (Java), using an object oriented approach to program design. Students develop their programming skills by constructing computer programs for a variety of applications. The course provides a foundation for all later courses in computer science, and develops programming skills useful for students in many other disciplines.

Learning Objectives

By the end of the course, students should be able to read, comprehend, design, and construct small programs using the Java programming language and an object-oriented design approach.

This objective contributes in particular to the BE graduate attributes 3(b) and 3(f) and the BSc (COMP) graduate attributes 1 and 2.

The course involves a substantial practical component in which you will construct a range of programs to develop your understanding of programming and program design. Many of the assignments involve constructing a simple version of a useful computer application.

The course does not assume any prior knowledge of programming, though students who have done some programming will find their prior knowledge helpful. Students who have done a lot of programming should take COMP112 instead.

Course Organisation details

Dates, Times, and Rooms: Lectures, Tutorials, and Laboratories

COMP 102 is a trimester 1 course. The trimester starts on 29 February. The examination period at the end of the course is 10 June - 29 June.

There is a weekly timetable on the course website showing the times of the lectures, labs, and tutorial.

There are two streams of lectures for COMP 102:

- **Stream 1** Lectures: Mon, Tue, Thu 14:10 - 15:00 in KKLT303
- **Stream 2** Lectures: Mon, Tue, Thu 16:10 - 17:00 in MCLT102

A schedule of lecture topics, readings, and assignment due dates is available online. Copies of the lecture slides will be distributed at the lecture, but will also be available via the schedule page.

Each student should sign up for and attend two weekly 1 hour lab sessions

- **Labs** in Cotton 219 and Cotton 238.
- **Lab A**: choose one of Thu 3-4, 4-5, 5-6 Fri 10-11, 11-12, 3-4
- **Lab B**: choose one of Mon 11-12, 12-1, 1-2, 3-4 Tue 11-12, 5-6

The first lab is on Thu/Fri in the first week of the trimester.

Note there are six options to choose from for each of the two lab sessions. See the online timetable for the times you can choose from. You must sign up for a lab session on the web: (at <https://student-sa.victoria.ac.nz/>) (also linked from the course home page).

To work on your assignments beyond the scheduled lab times, you may use any of the ECS computing labs on the second floor of Cotton at any time, unless they are booked for another class. You may also use your own computer.

For students who want to go over material again or are having difficulty getting on top of the ideas in the course, there will be an optional tutorial:

- **Optional Tutorial:** Wed 16:10-17:00, 77 Fairlie Tce, room 306 (FT77/306), starting in the third week.

We will run **help desks** Wed 12-1, 3-4, Thur 11-12, 12-1, CO242B, and an optional **lab for women students** Friday 12-1, CO238, starting in the third week.

Withdrawal

The last date for withdrawal from COMP 102 with entitlement to a refund of tuition fees is

- **Withdrawal with refund:** Friday 11 March.

The last date for withdrawal without being regarded as having failed the course is

- **Withdrawal without refund:** Friday 13 May

Later withdrawals need to be approved by the Associate Dean on the basis of special circumstances.

Textbook and other Materials

The **textbook** for COMP 102 is: *Java Foundations: Introduction to Program Design and Data Structures*, by Lewis, DePasquale, and Chase, 3rd Edition, published by Pearson (2010: ISBN 0132128810, 2013: 0133370461).

Note that course does not follow the textbook closely; the textbook is intended to be a resource and to provide you with explanations that will complement the lectures. The assigned textbook matches the course better than any other Java textbooks that we have seen, but other Java textbooks could also be a useful reference if you already have them. Note that the assigned textbook is also the current textbook for COMP103.

Students are not required to have their own computers, but it helps and resources are provided to make it easy for students work on the programming assignments on their own computers.

Workload

COMP 102 is a 15pt course and therefore has nominal total workload of 150 hours. In order to maintain satisfactory progress in COMP 102, you should plan on spending at least 10 hours per week on this course. A plausible and approximate breakdown for these hours would be:

- Lectures (and optional tutorial): 3-4 hours
- Reading and preparation: 1 hour
- Lab Sessions: 2 hours
- Further work on the assignment outside the lab session: 3-4 hours

Some students will require more time than this; other students who have done some programming may require less time than this, especially at the beginning of the course.

School and Staff

The School of Engineering and Computer Science is located on levels 2 and 3 of the Cotton building and levels 2 and 4 of the Alan McDiarmid building.

The undergraduate computing labs are mostly on level 2, though one lab is on level 1. The School office is on level 3: Cotton 358. The head of the School is Stuart Marshall, and the Dean of Engineering is Professor Dale Carnegie.

The staff for COMP 102 are

- Xiaoying Sharon Gao. ("Sharon", Course Organiser)
 - Cotton 333
 - +64 4 463 5978
 - Xiaoying.Gao@ecs.vuw.ac.nz
- Peter Andreae ("Pondy", lecturer)
 - Cotton 336
 - +64 4 463 5834
 - Peter.Andreae@ecs.vuw.ac.nz (or pondy@ecs.vuw.ac.nz)
- Roman Klapaukh. ("Roma", assignments and Wednesday tutorial)
 - Cotton 230
 - +64 4 463 5233 x 4034
 - roman.klapaukh@ecs.vuw.ac.nz

- Zarinah Amin (Senior Tutor)
 - Cotton 343
 - +64 4 463 5936
 - zarinah.amin@ecs.vuw.ac.nz
- Monique Damitio (Course Programmer)
 - Cotton 327
 - +64 4 463 5971
 - monique.damitio@ecs.vuw.ac.nz

Announcements and Communication

The main means of communication outside of lecture will be the COMP 102 web site at http://ecs.victoria.ac.nz/Courses/COMP102_2016T1/. There you will find, among other things, this document, the course [schedule](#) (with links to copies of the lecture slides), [assignment handouts](#), the [COMP 102 Forum](#), and the assignment submission system. The forum is a web-based bulletin board system. Questions, comments, and responses can be posted to the forum. Staff will read the forum posts and will frequently respond to them also. You should make a bookmark to the course home page because you will need to access it frequently.

Assignments and Practical Work.

There will be **10 weekly lab assignments**. The course web site has a page listing [the assignments, due dates, and resources](#). The lab assignments will be described in handouts that will be distributed at the Thursday lectures, and will also be available from the website.

The first lab assignment is mainly an introduction to using the computers in our lab facilities and the software that you will be using throughout the course. The other weekly assignments consist primarily of programming tasks. The assignments are a critical learning component of the course. They address most of the concepts and techniques that are introduced in the lectures, and they are the most important way of coming to grips with the material in the course. Due dates and how to submit your answers to the assignments is addressed below.

The assignments (except the first) are intended to take about 5 hours per week, but the actual time required will vary considerably from student to student. The scheduled lab sessions will help you get started the assignments, and then help you solve issues that you are stuck on, but **you will need to spend more time outside the scheduled sessions**. There will be scheduled help-desk times when a tutor is available to answer individual questions about the assignments, and we will also provide on-line helpdesk assistance.

Most of the assignments contain five parts:

- an **Exercises** part, to be completed in the lab, consisting of very small programs to ensure you understand how to use the key new programming constructs for the week. You will generally work on these in small groups, to get to know other students who you can study with and to develop your collaboration skills.
- a **Core** part, which you should aim to start in the first lab, and involves constructing programs using the basic concepts and constructs,
- a **Completion** part, which extends the Core part and involves more difficult problem solving and may cover additional concepts and constructs,
- a **Challenge** part, which usually involves substantially more difficult programming and may require you to find out new concepts and constructs by yourself.
- a **Reflection** part, which involves thinking and writing about your experience of the assignment.

The Exercises part is not assessed, though you need to do it. The Core part will be worth around 60% of the assignment, the Reflection part will be worth 10%, and the Completion and Challenge parts are the other 30%. The programs will be marked primarily on whether they work correctly, but there will be some weighting for good design.

We expect all students to be able to complete most of the Core and Reflection parts, and most to at least attempt the Completion part. The Completion part is important for students planning to go on to COMP103. The Challenge components are for students aiming for an A. If it takes you more than about 6 hours to complete the Core part of an assignment, we suggest that you should not spend additional time on the Challenge part. It is probably better to spend the additional time on the Reflection part, reading the textbook, going over your notes from the lectures, going to the tutorial, or working on questions and problems with other students.

Assignment Submission

Assignments are due

- **Assignments 2-5, 7-9:** at 1 pm on the Thursday of the week after the assignment was handed out.
- **Assignment 6:** 5 May (after the mid-trimester break)
- **Assignment 10:** due Fri 3 Jun (the last day of lectures).

When you have completed them, the assignments should be **submitted via the online submission system**. This means that you can submit assignments from the ECS labs or from a computer at home (or anywhere on the internet). You may resubmit as many times as you wish, *but the most recent submission of a file will be marked*. (You'll learn about how to submit assignments using a web browser in lab 1).

Assignment Marking and Late Penalties

All the assignments are important for your learning. All but the first assignment will be marked, and will together contribute a total of 20% to your final grade.

We will mark the assignments as quickly as possible; our goal is to have the marks and comments returned within one week of the submission time. Your marks and comments on your submission will be accessible via the web: see the links on the [Assignments](#) page.

Note that the **mandatory course requirement** is that you must submit reasonable attempts for at least seven of assignments 2 - 10. A "reasonable attempt" is a D (40%) or better. If you miss more than one of the assignments, contact the lecturer or the Senior Tutor as soon as possible. Students who have not met these mandatory requirements will be required to do make-up programming assignments in order to be able to pass the course.

Model solutions to the assignments are generally made available shortly after the assignment deadline (in the lecture following the deadline), so that you can review and assess your own work, and also build on the model solutions for the next assignment. Comparing your work to the provided solutions is an important part of the learning. This means that assignments submitted after the solutions are posted will **generally not be marked**, unless you have made arrangements on the basis of exceptional circumstances with the lecturer or senior tutor.

Group Work

We encourage you to work on the exercises in small groups in the labs.

In COMP 102 you are also permitted and encouraged to work on the core and completion parts of the assignments in pairs. (Note: this is special to COMP102 and is not generally true for Engineering and Computer Science courses!) **As long as you both include the name of the person you worked with** on your assignment, you and your partner may each submit the same answer for the Core and Completion parts. You must do the Challenge, and Reflection parts of the assignment yourself, and you may not work on shared code in groups of more than two. Make sure you read the section on plagiarism below.

You may choose a partner yourself, and you do not have to have the same partner (or any partner) for all the assignments. When choosing a partner, find someone with a similar level of confidence as yourself - working with a partner who is much more confident about programming than you will probably not help you to learn the material.

Getting Help: help desk and online help.

To help you when you are having difficulties with the assignments, we have tutors able to respond to queries via an on-line helpdesk. The online tutor will be responding to queries throughout the week, but we cannot guarantee an immediate reply. We also intend to have a tutor in one of the labs at certain times. When they have been determined, the times will be announced and posted on the course web site. .

We strongly advise you **NOT** to leave the assignments to the last minute, since there may not be any help available near the submission deadline. In particular, you should not have an expectation that online help will always be available late in the evening, especially the night before the deadline..

Tests and Exams

There will be two 45 minute in-term tests worth 15% each, held 6pm - 7pm on

- **Test #1:** Wednesday, 6 April, 6-7pm in COLT122, MCLT101, MCLT102, MCLT103, and
- **Test #2:** Wednesday, 11 May, 6-7pm in COLT122, HULT323, MCLT103, KKLT303.

You should contact the course organiser or the senior tutor as early as possible if you are not going to be able to attend a test at the scheduled time, or if you missed a test.

There will be an exam during the exam period.

- **Exam duration:** 2 hours

The [timetable for final examinations](#) will be available from the University web site at some time during the trimester. The final examination will be two hours long. No computers, programmable calculators or similar devices will be allowed in the test or final examination. Paper non-English to English dictionaries will be permitted.

All the assessment (assignments, tests, and exam) will address the learning objective of the course - reading, comprehending, designing, and writing programs in an object-oriented style. The tests and exam will assess all the material covered by the course up to the time of the test/exam.

The tests and the exam will be written on paper, not on the computer. While much of your learning will happen while working on the assignments at a computer, it is important to also prepare for the tests and exam by working on problems

on paper. All tests and exams from past years are available from the [Test/Exam Archive](#) on the website; you may use these to help you study for this year's tests and exam. Note that the structure of the tests and exam in previous years may be a little different, but they will still be excellent practice for you. Note that the exam this year will only be 2 hours, rather than the 3 hour exams in the past.

Grade Computation

Your grade for COMP 102 will be based on a combined mark for the assignments, the tests, and the exam:

Item	Weight
Assignments 2-10	20%
Test 1	15%
Test 2	15%
Final Examination	50%

If you get a higher mark in the exam than you did in a test, then we will boost your mark for the test up to your exam mark - we do not want to penalise students who took longer to get on top of the material, but got there in the end, as demonstrated in their exam.

Please note that that copies of their assessed work will be retained for inspection by the discipline review panel and the engineering accreditation panel.

Mandatory Requirements

The **mandatory requirements** for COMP102 are

- Obtain at least a D (40%) for at least 7 of the lab assignments 2 through 10.

The reason for the mandatory requirements is that the practical skills involved in being able to write and debug programs on a computer are an essential component of COMP 102, not tested by the tests and the exam.

Students who do not meet the mandatory requirements will receive a failing grade (K), even if they have high marks on the tests and exam. Students who have missed assignments should contact the lecturer as soon as possible to make alternative arrangements to meet the mandatory requirements.

Any student who is concerned that they have been (or might be) unable to meet any of the MCRs because of exceptional personal circumstances, should contact the course coordinator as soon as possible.

Passing COMP 102

To pass COMP 102, a student must satisfy the mandatory requirements and gain at least a **C-** grade overall.

Note: to be able to go on to COMP 103, you need to obtain at least a B- for COMP 102; just passing COMP 102 with a C-, C, or C+ will not be sufficient. Students planning on doing Engineering or Computer Science need to aim for a good grade in order to be able to get in to COMP 103.

Academic Integrity and Plagiarism.

Academic integrity means that university staff and students, in their teaching and learning are expected to treat others honestly, fairly and with respect at all times. It is not acceptable to mistreat academic, intellectual or creative work that has been done by other people by representing it as your own original work.

Academic integrity is important because it is the core value on which the University's learning, teaching and research activities are based. Victoria University's reputation for academic integrity adds value to your qualification.

The University defines plagiarism as presenting someone else's work as if it were your own, whether you mean to or not. "Someone else's work" means anything that is not your own idea. Even if it is presented in your own style, you must acknowledge your sources fully and appropriately. This includes:

- Material from books, journals or any other printed source
- The work of other students or staff
- Information from the internet
- Software programs and other electronic material
- Designs and ideas
- The organisation or structuring of any such material

Find out more about plagiarism, how to avoid it and penalties, on the University's website:
<http://www.victoria.ac.nz/home/study/plagiarism>

The [School policy on Plagiarism](#) is available from the course home page. Please read it. We will penalise anyone we find plagiarising, whether from students currently doing the course, or from other sources. Students who knowingly allow other students to copy their work may also be penalised. If you have had help from someone else (other than a tutor), it is always safe to state the help that you got. For example, if you had help from someone else in writing a component of your code, it is not plagiarism as long as you state (eg, as a comment in the code) who helped you in writing the method.

In COMP 102, we may use an automated system to check all submitted code in order to identify students submitting the same code who have not stated that they were working with a partner. The system does not make any code available to people other than the organiser of the course and the managers of the system.

Copyright: Although students own the copyright of any code that they write for an assignment, the COMP 102 assignments all build on some code that we have provided. This code is copyright by the university. We give you permission to use this code for your assignments, but you are not permitted to post to the web, or distribute in any other way, your answers if they include any of our code. Also, you may not distribute model solutions to anyone other than a student currently in the course. Breaking this rule is a violation of the university's statute on student conduct. If we find that you have distributed our code on the web or by other means, we will seek disciplinary penalties. **Don't ruin the course for future students by giving them answers!**

Rules & Policies

Find key dates, explanations of grades and other useful information at <http://www.victoria.ac.nz/home/study>.

Find out about academic progress and restricted enrolment at <http://www.victoria.ac.nz/home/study/academic-progress>.

The University's statutes and policies are available at <http://www.victoria.ac.nz/home/about/policy>, except qualification statutes, which are available via the Calendar webpage at <http://www.victoria.ac.nz/home/study/calendar> (See Section C).

Further information about the University's academic processes can be found on the website of the Assistant Vice-Chancellor (Academic) at <http://www.victoria.ac.nz/home/about/avcacademic>

All students are expected to be familiar with the following regulations and policies, which are available from the school web site:

[Grievances](#)

[Student and Staff Conduct](#)

[Meeting the Needs of Students with Disabilities](#)

[Student Support](#)

[Academic Integrity and Plagiarism](#)

[Dates and Deadlines including Withdrawal dates](#)

[School Laboratory Hours and Rules](#)

[Printing Allocations](#)

[Expectations of Students in ECS courses](#)

The School of Engineering and Computer Science strives to anticipate all problems associated with its courses, laboratories and equipment. We hope you will find that your courses meet your expectations of a quality learning experience.

If you think we have overlooked something or would like to make a suggestion feel free to talk to your course organiser or lecturer.

[Course Outline as PDF](#)
