1. Course Overview & Report Context

COMP1100 is the core first-year programming course in RSCS, and aims to introduce the “art of programming in small steps”\(^1\). It is a compulsory course for the Bachelor of Advanced Computing (BAC) and Software engineering (SEng) degrees, and also optional (with COMP1130 Art of Computing) for the Bachelor of Information Technology (BIT).

COMP1130 is a variation on COMP1100, which aims to extend the material covered in 1100 by providing “links into current research as well as to provide ‘behind the scenes’ perspectives and a more foundational approach to the material”\(^2\). It is a compulsory course for the Bachelor of Computing R&D (BAC R&D) and is an optional replacement for COMP1100 in other degree programs.

Overall enrolments for COMP1100 and especially COMP1130 have been increasing in the last few years, although the total number of enrolments is similar to previous peak years in 2002 and 2010 as shown in figure 1.

The context for this report is a gradual increase in the failure rate since 2010 (see figure 2) and a decrease in student satisfaction (as measured by SELS) these courses in S1 2016, especially in COMP1130 (see figure 3).

Except where otherwise indicated, when we talk about “the students” in this report we refer to the students from both COMP1100 and COMP1130 in 2016.

2. Significant changes between 2015 & 2016

As well as the change of lecturer from Uwe Zimmer to Tony Hosking, the primary change from 2015–2016 was the move to using GitLab for all course materials and assessment. Previously, course materials had been maintained on a separate course website, but this approach was deprecated by CECS IT in the lead-up to the S1 2016 course. This required a significant amount of work to re-tool the labs & assignment submission system, and this work was the responsibility of the course lecturer and tutors.

\(^1\)http://programsandcourses.anu.edu.au/course/COMP1100
\(^2\)http://programsandcourses.anu.edu.au/course/COMP1130
Figure 1: enrolments (including students who withdrew/failed) from 1999–2016 (COMP1130 started in 2006. The two previous “peaks” represent the dot-com bubble (early 2000s) and when COMP1100 was removed as a compulsory part of the engineering degree after 2010.

Figure 2: overall failure rate (includes withdrawals) for COMP1100 and COMP1130 from 1999–2016 (dotted line indicates 5-year moving average).

There were a few other differences between the 2016 incarnation of the course and previous versions (listed in no particular order).

1. For the reasons described above, there was no full course outline available to students at the start of the course; the main COMP1100 page on the GitLab wiki “emerged” gradually over the semester.

2. Assignment 1 contained bugs/compiler warnings when it was released to students due to time pressures in moving the course material to GitLab.

3. Since Uwe’s lecture slides & presentation are highly specialised to his style of delivery, Tony re-wrote all the lecture materials for the course. These followed textbook more closely, so the order in which some things were was different to previous years.

4. The labs started one week later (week 2 vs week 1) because the CSIT labs were unavailable in week 1 due to ongoing refurbishments.
5. Students didn’t fit in the lecture theatre for the first three weeks, so Tony gave double-up lectures in these weeks. As a result, the COMP1130 content didn’t start until week 4. This problem was only “solved” when enough students stopped attending lectures.

3. Learning outcomes & assessment

In 2016, the assessment items for COMP1100/COMP1130 were:

- **lab assessment** in week 5 (5 marks)
- **assignment 1** released week 5, due week 8 (15 marks)
- **assignment 2** released week 8, due week 12 (15 marks)
- **mid-term exam** in week 7 (10 marks)
- **final exam** during exam period (55 marks, 40% hurdle mark)

The learning outcomes (LOs) for COMP1100 were:

1. describe and contrast various algorithmic components, and relate them to specific corresponding data structures
2. analyse program fragments and explain their types, behaviours and effects
3. design algorithmic solutions to simple problems and present their implementation in a functional programming language
4. design data structures appropriate for the representation and implementation of simple algorithmic problems
5. analyse small programming problems to suggest, design and implement appropriate modular program structures, based on a data-directed approach
In addition to these, COMP1130 had the following LOs:

1. understand proof by induction, recursion, the running time of programs, and the Big-Oh notation
2. understand various data structures including lists and trees and their associated algorithms, and be able to appropriately use these data structures in programs
3. understand how data structures and algorithms are employed in various research areas in computer science
4. have some appreciation of current issues in computer science research

COMP1100 and COMP1130 students were “assessed on the same basis, though [COMP1130 students] will be assessed on extended material (e.g., via extra questions, and extended requirements for assignments)”.

Each assessment item covered all of the LOs for the course, with the COMP1130-only LOs assessed through extra questions on the assignments and different exams to the COMP1100 students.

4. Feedback from students during the course

Feedback was sought from students through both class reps and feedback from tutors. One of the 2016 COMP1100 class reps independently conducted a “student satisfaction” survey during the course. This survey was initiated, designed and conducted by the class reps through Piazza. This survey didn’t indicate whether the student was in 1100 or 1130. See figure 10 for a full list of the questions and answers, and section 7.1 for a discussion of the results.

5. Review methodology

The review panel collected quantitative data from FAIS and qualitative feedback from students. All the data and associated analysis & visualisation code for this review is available on GitLab.

5.1. Quantitative data

- aggregated grade data for the period 1999–2016 (figure 9)
- all individual student scores & grades for the period 2012–2016
- an unenrollment timeline (extracted from FAIS logs) for the period 2012–2016 (figure 6)
- SELS results for the period 2012–2016 (figure 11)
- the results of a S1 2016 student-run mid-semester Piazza questionnaire which gives more detailed satisfaction data than the SELS results (figure 10)

2https://gitlab.cecs.anu.edu.au/u2548636/comp-1100-2016-peer-review
5.2. Qualitative data

- 60 student responses to a personalised email from Ben to all 2016 students (the full text of this email is provided in appendix A)

- 40 student responses to a similar personalised email from Ben to all 2009–2015 students (this was only sent on Aug 30, and results are still coming in)

- six in-person interviews with students, three from this year’s course (one aced it, two who failed), and three students who took the course in previous years

- one submission to the panel from Henry Gardner about a discussion he had with a parent of a bright student who struggled in COMP1130 in 2016 due to a lack of scaffolding

To get detailed qualitative feedback, we sent a personal email (from Ben, wearing his “student experience” hat) to all 2016 students instead of conducting focus groups because we felt that the students would be more likely to be honest in a private email discussion than in a focus group setting. The email did ask specific questions (see section A), but was also worded in such a way as to give the students the freedom to say (and even rant about) their experience in the course. Since one of the main aims of this review is to explain declining satisfaction and engagement rates, we decided that giving the students the freedom to rant in this way gave us the best chance of getting to the heart of their reasons for disengaging.

The SI Piazza questionnaire data was volunteered by one of the COMP1100 students (a class rep) as a result of getting this email from Ben.

6. Results

Only a subset of the data collected during the review process is presented in this report—more data and analyses can be produced on request.

6.1. Enrolment numbers

As shown in figure 1, although there was a significant uptick in enrolments this year (with associated difficulties), enrolments are about equal to previous peak enrolment periods in 2002 and 2010. However, if the growth continues, we will hit an all-time high for enrolments next year (and CS enrolment figures from the current Fall US semester indicate that the strong growth in CS enrolments will continue).

Figure 4 shows the enrolment numbers broken down by degree program. This figure clearly shows the drop-off in the “other” category due to COMP1100 no longer being compulsory for engineers.

6.2. Student results

Figure 2 shows the course failure rate since 1999 (the FAIS data only goes back to 1999). The 5-year moving average has been increasing since 2008.
Figure 4: Enrolments by degree program from 2009–2016.

Figure 9 shows that in COMP1110 in 2016 the majority of the year-on-year increase in the failure rate comes from an increase in NCN grades, although there is an increase in W (WD/WN) grades as well. Proportionally, there are still fewer NCNs than in 2013, but there are more Ws and more failing grades in total. The distribution of other grades is fairly stable over this time.

In COMP1130 the increase in failing grades is less pronounced, and the NCN rate is stable from 2015–2016. There is an increase in W grades, and a significant drop in HD grades. Figure 9b shows how COMP1130 has changed over the past five years from a small group of mostly D & HD students to a larger group (approximately 100 students in 2016) with a similar overall grade distribution to COMP1110.

Figure 5: Enrolments by degree program from 2009–2016.

Figure 5 shows the failure rate broken down by degree program. The large fluctuations in some of the subfigures are caused by small enrolment numbers, especially for COMP1130 (e.g. there was only 1 student in COMP1130 in 2016).
6.3. Unenrollment timelines

Figure 6: unenrollment timeline for COMP1100 and COMP1130 from 2012–2016.

Figure 6 shows the unenrollment timeline since 2012 (extracted from FAIS logs). The timelines are similar for COMP1100 over this time, with peaks in the first week and before census date. For COMP1130 in 2016, several students unenrolled in the first week, similar to COMP1130 but dissimilar to previous years in COMP1130.

7. COMP1130 SELS results

Figure 11 shows the COMP1130 SELS breakdown for all six questions from 2012–2016 (SELS data is only available from 2012 onwards). The 2016 results are worse than 2014 and especially 2015, and are similar to the 2013 results. The most negative results in 2016 were in response to the question: “the feedback I received during the course supported my learning”.

7.1. Mid-semester satisfaction survey

Figure 10 shows the results of the mid-semester Piazza survey conducted by the COMP1100 class reps, which received around 90 responses. Unfortunately, whether a student was in COMP1100 or COMP1130 was not recorded in this survey.

The questions pertaining to the assessment, the lectures and overall satisfaction all received more negative responses than positive, with the “how satisfied are you with this course so far” question receiving 46% negative, 32% positive and 22% neutral.

Interestingly, this survey also asked students about their previous programming experience (none, some or a lot). Figure 7 shows the responses to a few of the questions broken down by programming experience. Overall, students with a lot of programming experience were the
most dissatisfied with the lectures. In terms of overall satisfaction, students with no previous programming experience were the most dissatisfied (59% negative 17% positive), followed by those with a lot of programming experience (40% negative 47% positive), followed by those with some previous experience (27% negative 50% positive).

This survey also contained a free-text comment section. The main themes to emerge from these comments are the same as those from the feedback emails (see section 7.2): insufficient scaffolding and support for beginner programmers, frustration with the course content being spread across different websites (e.g. Piazza, Wattle, GitLab) and not available earlier, the steep learning curve, and insufficient support from tutors in the labs.

7.2. Student feedback emails & interviews

The personalised email from Ben received around 60 responses from 2016 students, with a reasonable spread across the different grade cohorts. There were also around 40 responses from students from previous years.

Discerning overall trends from these individual responses is difficult—the students are a diverse bunch, and had significantly different experiences in the course. Unsurprisingly, some students were extremely positive, others were extremely negative, and a whole range of opinions in-between. However, there were some recurring comments/themes.

1. Students without prior programming experience (even bright students) found the course too difficult, and in particular that the course lacked “scaffolding”—they spent a lot of time learning incidental skills not covered in the course (e.g. how to set up their programming environment, how to use GitLab) before even engaging with the course content.

2. Students experienced a steep difficulty curve and misleading feedback—many students thought they were doing ok with the early labs, but then bombed on the mid-semester exam or first assignment. Students expressed a desire for more (and earlier) feedback through assessment items.

3. Students were frustrated by the course material not being released earlier, and also being confusingly spread across several different websites (Piazza, Wattle, GitLab).

4. Students who were struggling found some (although not all) of the tutors dismissive and the labs discouraging. Many reported feeling like outsiders in CS. A “shadow network” of alternate COMP100 labs emerged for Chinese international students.

5. Many students had an expectation of learning “applied” computing skills, and complained that the course is too theoretical and lacked “real world” applicability (e.g. complaints about Haskell).

6. Amongst COMP130 students (especially those who didn’t do well), there were complaints about a lack of cohesion between the content of the lectures, labs and assessment items.
Figure 7: 2016 COMP1100/1130 Mid-semester student satisfaction survey results broken down by (self-reported) prior programming experience.
One of the broader points discussed at length by the review panel is that the COMP1100 student body is extremely diverse. From the student feedback emails and our experience in teaching this course, in our opinion two useful axes of differentiation are mathematical ability (not previous programming experience) and CS “ambition”—whether the student is a CS student, or just looking to gain useful computing skills for application in other domains. Figure 8 shows these axes graphically, with the four “quadrants” enumerated. As an example, an actuarial studies student may be in quadrant 1, while a BAC student with limited maths background might be in quadrant 4.

Placing individual students on the graph is difficult, especially since this is a first-year, first-semester course and we cannot look at a student’s performance in previous semesters. The panel requested detailed “high school maths level” data from Central, but this data is time-consuming to collate and was unavailable at the time of writing this review. As a result, we have not performed any of the quantitative analyses using this breakdown—although we are keen to do this in the future if we can get the appropriate metadata. Instead, we include figure 8 to ensure that we consider the ramifications for students in all of these quadrants of each of the recommendations in this report.

8. Recommendations

To address the withdrawal and failure rate amongst COMP1100 students this panel recommends that in 2017, COMP1100

1. is given resources (e.g. tutor budget) to have lectures labs, & assignments ready, clearly organised and presented to students at the start of semester
2. provides clear documentation showing students how to set up and use the software environment (e.g. text editor, Haskell/Stack, GitLab) required to participate in the course using their own machines

3. has a plan for conducting lectures with 450+ students in S1 2017 (e.g. holding lectures in Manning Clark Theatre 1)

4. has tutors for S1 2017 finalised in November 2016 with induction and training conducted well in advance of the start of S1 2017

5. employs a senior tutor to provide cultural oversight over the tutors and lab experience

6. is given resources (e.g. IT support) to set up and maintain scalable infrastructure for providing course content, assignment submission, (semi-automated) marking and feedback to students

7. includes an early diagnostic hurdle assessment with a graceful exit into COMP1030 (Art of Computing) for students who don’t get through

8. includes more early assessment to provide a more gradual difficulty curve and to give better feedback to students about their progress

9. include more face-time between students and instructors, e.g. lower tutor-to-student ratio in labs, or tighter integration between labs & PAL sessions

In addition to these recommendations, to address the declining satisfaction rate amongst COMP1130 students this panel recommends that in 2017, COMP1130

10. includes clear communication in week one about COMP1030, COMP1100 and COMP1130—what they are, what they aren’t, and who should do which—and reviews study plans on the programs and courses website.

11. clearly communicates to students how the extra COMP1130 content fits into the big picture of the rest of the course

12. provides completely different assessment items for COMP1130, instead of “COMP1100 assessments plus extensions”.

Finally, this panel recommends that the RSCS continues to re-think issues surrounding our first-year programming courses, including the need for a “prog zero” course which lays the foundation for COMP1100, and a clearer picture of which computing course (Art? Programming for Scientists?) the students in quadrants 1 & 2 should take.
(a) grade breakdowns from 1999–2016 (note the y-axis scale is different for COMP1100 and COMP1130)

(b) grade breakdowns (as a proportion) from 1999–2016.

Figure 9
Overall, how satisfied are you with the course so far? (n=87)

If you attend them, how satisfied are you with the PAL peer assisted learning sessions so far? (n=55)

How satisfied are you with the Piazza site as it is used in this course i.e. ease of use and ability to find content? (n=87)

How satisfied are you with the GitLab site as it is used in this course i.e. ease of use and ability to find content? (n=88)

If you have used them, how satisfied are you with the Echo360 recordings of lectures so far? (n=62)

How satisfied are you with the lectures so far i.e. the pace of content, ease of understanding and quality of teaching? (n=88)

How satisfied are you with the labs so far i.e. the pace of content, ease of understanding and quality of teaching? (n=88)

If you have asked outside class for feedback or assistance (i.e. through Piazza or email) how satisfied were you with the response? (n=69)

How satisfied are you with the way assessments are conducted in the course so far? (n=88)

How satisfied are you with the level of assistance and feedback in lectures and labs so far? (n=87)

How satisfied are you with the level of assistance and feedback in lectures and labs so far? (n=87)

How satisfied are you with the PAL peer assisted learning sessions so far? (n=55)

Overall, how satisfied are you with the course so far? (n=87)
A. Student survey email

Hi <name>

My name is Ben Swift, I’m one of the lecturers in the ANU Research School of Computer Science. I’m also the “student experience” convenor for the school, so it’s my job to listen to students about what parts of their experience are good, and what parts aren’t.

We’ve already started thinking about how we can make COMP{ččČČ,ččďČ} better next year, and since you took that course in S1 this year I was hoping you’d be willing to give me some feedback.

(W/NCN/N students only)

I realise that you might have had a rough time in this course, so I wanted to reach out and see what we can do better.

This isn’t a big, formal questionnaire, I just want to give you a chance to have your say on what worked (and what didn’t) in COMP{1100,1130}.

To answer, just reply to this email. You can tell me whatever you like, but if you want some talking points then there are a few specific questions at the end of this email (just answer inline). If you don’t have anything specific to say about a given question that’s fine too.

Thanks for helping out, and this feedback won’t be used against you in any way. Feel free to be honest (rants are ok), because I value your feedback and want to make our courses better. If you’d rather chat in person, just let me know by reply email and we can tee something up.

Cheers,
Ben

• Why did you take COMP{1100,1130}?

• At the end of the course, did it live up to these expectations? Why/why not?

• In general, did the labs work for you? What about the peer-assisted learning (PAL) sessions? If not, why not?

• Would it have been useful to have some warm-up training (e.g. an optional short course) before the course, and what topic(s) would have been most helpful?

• Did you start out in 1130 and switch to 1100? Why? (COMP1100 students only)

• Why did you decide to take 1130 instead of 1100? (COMP1130 students only)

• What’s the main reason you didn’t get through the course? (W/NCN/N students only)

• Did you know about the deadline for withdrawing for the course without failing? (WN/NCN students only)

• After {1100,1130}, have you continued on with computer science, or have you switched to something else?
• Any other comments? Anything you want to get off your chest? Suggestions for the future?
(a) I had a clear idea of what was expected of me in this course

(b) The teaching and learning activities (e.g., lectures, tutorials, field trips) supported my learning

(c) I had ready access to the learning opportunities provided in this course (e.g., course notes, online materials, library resources, field trips)

(d) The assessment seemed appropriate given the goals of the course

(e) The feedback I received during the course supported my learning

(f) Overall, I was satisfied with my learning experience in this course