Is comprehension or application the more important skill for first-year computer science students?

Nickolas J. G. Falkner[†]
School of Computer Science, University of Adelaide

Abstract

Time and performance data was collected on a class of 147 Computer Science 1B students, where students carried out a design and programming task based on one that had been seen in a previous examination. Given that students had previously worked through the task, we assessed their comprehension of that material in this assignment. We were then able to collect the performance data and correlate this with the examination marks for the student to determine if there was a relationship between performance in the examination and performance in this practical. We were also able to correlate the performance in this practical with the time taken to complete the practical, and with the student's statement as to whether they remembered how they had solved it in their previous attempt. By doing this, we discovered that the students who remembered having solved it previously had a significantly higher mean examination mark than those students who claimed not to remember it. Unsurprisingly, students also performed better in this assignment if they had performed better in the examination. The mean time to complete the task was significantly less for those students who claimed to remember the task. In this task, the comprehension of the original material and the ability to recall it was of more importance than the ability to apply knowledge to an unseen problem.

Introduction

The education of computer science students is, relative to many other disciplines, a new pursuit carried out on a new technology. Arguments over fundamental educational approaches, such as the mathematically-oriented and tool free approach proposed by Dijkstra (1989) versus 'traditional' tool and technique based courses (Clear 2006), still take place as do detailed, and sometimes heated, discussions as to which tools should be used. While computer programming is not the whole of computer science, any more than a telescope is the sum of astronomy (to borrow from Dijkstra), it is an important component of the instruction that students receive. Computer science students need to be competent programmers once they graduate and must be able to demonstrate a knowledge of skills and techniques, with the ability to apply them in different circumstances and, ultimately, judge which is the best approach to use in a given situation.

The main first year course in computer science at the University of Adelaide provides a basis of computer programming in the *Java* programming language and builds to provide a set of tools that a student can choose from to solve problems. The year is divided into two semester-long courses, Computer Science 1A (CS1A) and Computer Science 1B (CS1B). Computer

Science 1A is a pre-requisite for computer science 1B and the two may not be taken at the same time. Course assessment consists of practical work, tutorial work and examination. The practical work mostly takes the form of software design and programming, in a programming laboratory environment. The practical work is not only a tool for assessment but it is one of the many ways that students learn to interact with other students and staff -we consider this to be an important skill for all of our students to develop (University of Cambridge 2007). The practical work is used to reinforce the points raised in lectures and to develop the vital skill of programming that the students will need for the rest of their university and professional career. This is an application of a practical skill relevant to our discipline and students are provided with notes, guides and practical demonstrator support to achieve their goals (Cannon & Newble 2006). Assessment dominates the way that our students learn: the practical assessments that are provided in computer science are one of the most dominating features of ongoing assessment, due to the amount of time and level of attention required (Ehmann 2005). Additional courses in first-year computer science include Internet Computing (2008) and, from 2009, Puzzle-Based Learning.

Having introduced programming concepts in CS1A, we should then be able to build on a student's understanding of these concepts in later subjects to allow him or her to, eventually, become capable of making high-level decisions regarding choices of programming languages and techniques, rather than just making low-level decisions on particular programming constructs in a given language. It is not enough that a student knows that a concept exists, they must understand how it is used, how it fits in with other concepts, how it can be combined with other concepts and, ultimately, how to fashion new knowledge that builds on their existing understanding. The foundation is key, however, and where a problem can be identified, it must be addressed as quickly as possible or a student's computer science knowledge may be weakened, or contain substantial holes.

The 2007 semester 1 CS1A examination contained a question dealing with the concept of inheritance in the *Java* programming language (Horstmann 2008). This question attracted relatively lower marks than other questions in the paper. Inheritance is a very important concept for further study in computer science and it was decided that a new practical assignment would be added to CS1B in the subsequent semester to provide practice for students, to develop their foundation knowledge. From a teaching perspective, it is essential that students not just be able to pass examinations, but that they have an understanding of the subject matter, which in turn then allows them to pass the examinations. To provide a mechanism for understanding and explaining the different stages of learning that we are addressing we used Bloom's taxonomy (Bloom 1956) as a framework. The original examination, the formation of the subsequent assignment and the learning outcomes from this project were developed in accordance with local guidelines for good teaching practice and shared experience. Bloom's taxonomy provides us with a structure for shared discussion beyond our discipline.

Bloom's taxonomy divides the cognitive domain into six taxa: knowledge, comprehension, application, analysis, synthesis and evaluation. Our computer science 1A course is designed to take students from a state of subject ignorance up to a point where they can comprehend the knowledge that they have gained and apply it to a new situation. For example, having learnt how to program in *Java*, they are expected to be able to solve unseen programming problems in the examination. The examination question on inheritance contained tests of comprehension and application and, as it was executed poorly, indicated that a number of students had difficulty in one or both of comprehension and application.

Work conducted since Bloom has shown that prior knowledge is still the most important contributor to learning but that it is the availability of that knowledge, often through some form of cueing, that is the most significant contributor to learning and development (Dochy, de Rijdt & Dyck 2002). Dochy et al. also recommend that assessment be refined to take prior knowledge into account and that student learning can be greatly assisted by feedback on the quality of a student's solution. The new practical was designed to allow students who had previous knowledge, and had recall of that previous knowledge, to demonstrate their comprehension. We also provided demonstrator support to provide immediate feedback on the solution to reinforce student learning. This also increased the involvement of the students in their teaching environment, a significant contributor to the student's academic skill development (Terenzini, Theophilides & Lorang 1984).

Our measurement of comprehension was defined as a combined measure of the student's recollection of the material, combined with a measurement of their time taken to perform the assignment. Thus, this practical assignment could also be used to gather data on how long all students took to perform the task and could also be used to give the students awareness of their own software development process. The practical assignment had the following mutually-supportive goals:

- To give students an opportunity to attempt the problematic examination question again in a supervised practical environment. This allows students to demonstrate knowledge and comprehension.
- To provide students with a guide as to the time that they took to perform each stage of the answer process. This is part of the feedback process.
- To give students immediate feedback on the design process that they employed.
- To provide timing data to staff on the student process. This allowed us to measure the differences in comprehension across the class.
- To provide a feedback opportunity for students on the assignment.

Given that some students perform better than others in examinations and programming assignments, we wished to determine whether it was the student's ability to comprehend the work that was significant or whether it was their ability to apply their comprehension. In this context, a better student achieves either a very good result, or a good result in a short time. This also highlights a fundamental distinction between students, where an academically inclined student will move more quickly to a level of application and theorising versus a non-academically inclined student who will slowly move up from note taking and requires more time and active involvement to move to the theoretical level (Biggs 1999).

In order to answer this this question, we suggested a range of hypotheses:

1. Students who remembered, or thought that they remembered, how they had solved the problem before would perform better in this assignment. This was a test of comprehension, based on existing knowledge, within the framework of Bloom's taxonomy (Bloom 1956). A student who has comprehension can successfully identify items of knowledge that they have previously seen and would, therefore, be more likely to recognise the exam question and potentially recall the solution. This would also be a reflection of the availability of this knowledge to the student (Dochy et al. 2002).
2. Better students, as defined by their previous examination performance, would take less time to solve the problem. A better student would already be functioning at a higher level, that of application, and would be able to solve a new problem more quickly anyway. A student capable of application can solve new problems and use the knowledge that they comprehend in a new situation (Bloom 1956).

3. Students who perform better in examinations also tend to perform better in assignments. This is another aspect of application but also involves such aspects as successful time management, to bring sufficient resources to bear to successfully complete a task.

Methodology

The examination question from the 2007 Semester 1 CS1A examination was incorporated into a student booklet that contained detailed rubrics and blank boxes for students to fill in the time as they carried out their task. The data collected from each student consisted of:

- Their student number, practical session day and time.
- Timecheck 0 - the time at which they started.
- Timecheck 1 - the time at which they finished reading the question.
- An indication as to whether they remembered how they had solved it originally.
- Timecheck 2 - the time at which they had finished part 1.
- Timecheck 3 - the time at which they had finished part 2a and 2b.
- Timecheck 4 - the time at which they had finished part 2c.
- Timecheck 5 - the time at which they had finished part 3.
- Coding timechecks - times as they finished each stage of code production.
- Anonymous feedback - a tear-off sheet with feedback could be separated from the booklet and handed back with no student number information.

Practical demonstrators were also given a booklet that contained detailed instructions, solutions and a marking scheme. To narrow the range of possible marks, students were rated on each part with a rating of Incomplete, OK or Good, based on a set of marking criteria. These ratings were converted to a score of 0, 1, 2 or 3, for four separately marked parts, with a total mark out of 12. Demonstrators also provided written and verbal feedback to students on the way that they solved the problem.

Students were not shown the booklet before attending the practical and they were not allowed to retain them on completion. The assignment took place in the computing laboratories, with extra practical demonstrators to provide assistance. Students were allowed to use their notes or textbook if they wished. Attendance at the start of the session was compulsory but student participation was voluntary, as the assignment was worth no marks due to its formative nature. Students who left without any other data entry were asked to make a comment on the feedback form to explain their reason for leaving.

During the practical assignment students read through the book, wrote answers to questions and entered their timing data. If, at some later stage, they wished to return to a previous part of the assignment and carry out more work, they were asked to write an 'X' next to the original timing information to indicate that more time was being spent. This was considered to be the easiest way to reflect that the time taken in a given section was no longer in keeping with what was recorded without imposing a complex time updating scheme on the students. Simply viewing a previous part did not require any action on the part of the student.

The nature of self-reporting of timing data is that it is possible for a student to fabricate their times in an attempt to either reduce the amount of time spent, if they feel that they have spent too long, or to increase the time spent, if they felt that they had rushed. Rather than ask students to report a set number of minutes, we asked them to write down the clock time at each time check. This reduced the likelihood of fabrication as students then couldn't see an explicit number of minutes written down on each page, and were less likely to go to the effort of determining a more desirable time and calculating the appropriate second clock time. The granularity of the timing data was sufficiently coarse that variability in reporting would be reduced over the number of time checks completed, in that all time checks would have to be varying in the same way to obtain a substantial deviation.

On completion, the booklets and feedback sheets were separated and returned to the author. After collating the data from students, the booklets were returned to students to allow them to consult their own work and the demonstrator's comments for improvement.

Results and Discussion

147 students undertook the activity and returned their books. The mean CS1A examination mark for this group was 69%, with a standard deviation of 13 percentage points. While this may seem high, students cannot enter CS1B unless they have passed CS1A and the left-hand tail of the examination distribution is cut at 50%.

The data was cleaned to remove incomplete data, where students had failed to fill in boxes or had provided impossible data, and data where it was impossible to determine the total time spent in the phases. Students who had marked sections with an 'X' may still have provided valid data for start and end time and their data could be used. This reduced the dataset to 99 students, with an mean CS1A examination mark of 72%, standard deviation of 13 percentage points. This dataset was identified as 'Cleaned'. The coding timechecks were separated from the development timing data and are not analysed further in this paper, as they have no bearing on this study.

The practical script contained the following question:
"Do you remember how you solved [the attached question] during the exam? Please circle one of the answers below. If you didn't sit CS1A last semester, circle NO.".

The dataset was further divided into two groups, those students who had said that they did not remember the answer to the question, and those students who claimed that they did. The question was designed to be simple and to have a yes or no answer, rather than giving students a Likert scale based on their impression of how much they remembered. Thus, it is possible that a number of students thought they knew more than they did and a number of students thought that they recalled less than they did.

The first group formed the dataset 'N Only' and the second group formed the dataset 'Y Only'. Table 1 shows the mean CS1A mark and time taken for each of the score bands for the 'N Only' dataset. Table 2 shows the same information for the 'Y Only' dataset. The possible marks for the assignment were out of 12 and the student results were grouped into two-mark bands, 11-12, 9-10, 7-8 and 5-6.

Band	Mean CS1A Mark	Mean Time
11-12	69%	0:44
9-10	70%	0:44
7-8	66%	0:39
5-6	62%	0:48
	n = 66	

Table 1: Mark band versus mean data for N Only

For the 'N Only' dataset, the mean CS1A mark was 68%, with a standard deviation of 12 percentage points. The mean result for the assignment was 9.5, with a standard deviation of 1.6.

Band	Mean CS1A Mark	Mean Time
11-12	84%	0:32
9-10	69%	0:37
7-8	68%	0:37
5-6	No entries	
	n = 33	

Table 2: Mark band versus mean data for Y Only

For the 'Y Only' dataset, the mean CS1A mark was 79%, with a standard deviation of 11 percentage points. The mean result for the assignment was 10.7, with a standard deviation of 1.4.

The first interesting item is the significant increase in mean CS1A mark for those students who successfully read and applied the instructions, and were able to solve the problem in a methodical manner without returning to previous work. This assignment was based on an examination question that was designed to be undertaken in a controlled environment under examination conditions. The two hours allocated for the assignment was exceedingly generous. A student who had successfully recalled information on inheritance, and had comprehended this information, should have no difficulty in answering the question but, more importantly, should be operating at a level where they did not have to revise their work in the face of new questions. Students were encouraged to read through the questions carefully before beginning. Despite this, 57 students had to revisit a previous part at least once. In some cases, students revisited previous parts and made changes as many as eight times.

Removing those students who forgot to enter timing information or could not answer the questions in one pass raised the mean mark by one standard deviation to 72%. On examination of the removed data, none of the removed students had achieved High Distinctions (although several of the remaining students had) and only 5 students had achieved Distinctions. Questioning of the student body after the assignment revealed that a number of students had not filled in all of the boxes because they didn't see the importance of filling in each one, despite the detailed instructions in the booklet and verbal instructions delivered at the start of every session. The most common reason given for this omission was that the assignment "wasn't worth any marks".

Figure 1 shows the graph of CS1A mark (%) versus the time spent in the different phases during the assignment for the entire Cleaned dataset. The highly dispersed nature of this data, with a potential pattern but of a highly diffuse nature and a correspondingly poor correlation with a linear trendline of 0.15481, led us to further classify the data into a set of mark bands. These mark bands allowed us to see the larger structure of the data, which had originally been masked by its point cloud nature. Figure 2 shows the CS1A score versus the band, with a linear trendline, for the Cleaned dataset. The R2 value for the trendline is 0.9799, indicating a high degree of correlation.

Although there is some indication of accretion in Figure 1, there is no strong indicator of correlation apart from an observation that students who perform less well in their examination take longer to execute the assignment. However, there is a strong correlation between the success in the examination and the final mark achieved for this assignment, as shown in Figure 2.

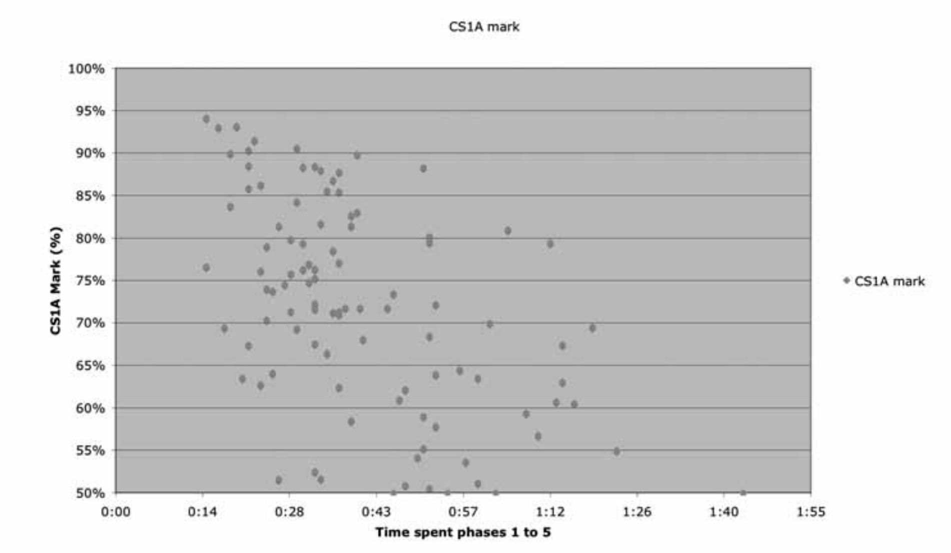

Figure 1: CS1A mark versus time spent in phase

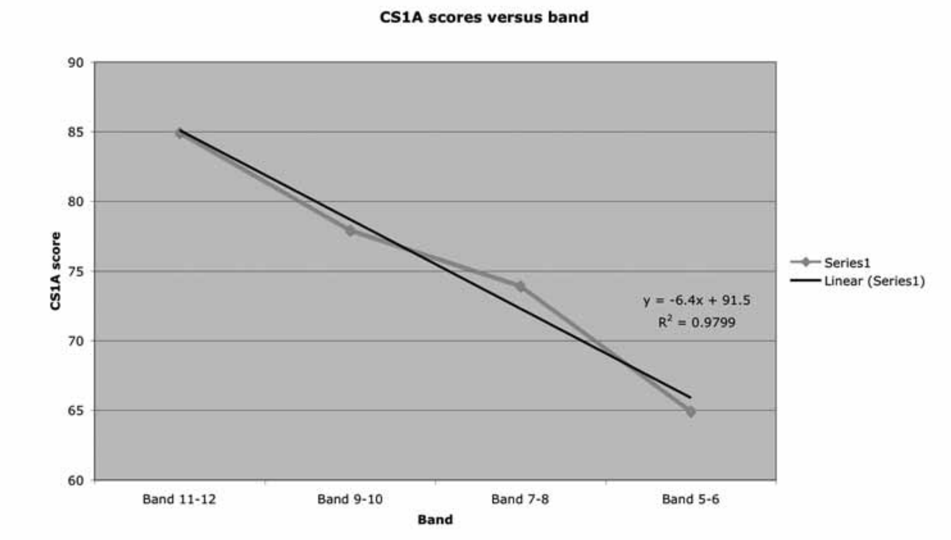

Figure 2: CS1A scores versus band, with trendline

The next three charts, Figures 3, 4 and 5, show the final mark for the assignment versus the time spent to complete it. The first shows the Cleaned dataset, but with separate markers for the Y and N datasets, 4 shows the graph for the 'Y Only' dataset and 5 shows the graph for the 'N Only' dataset. There are several interesting points in this data.

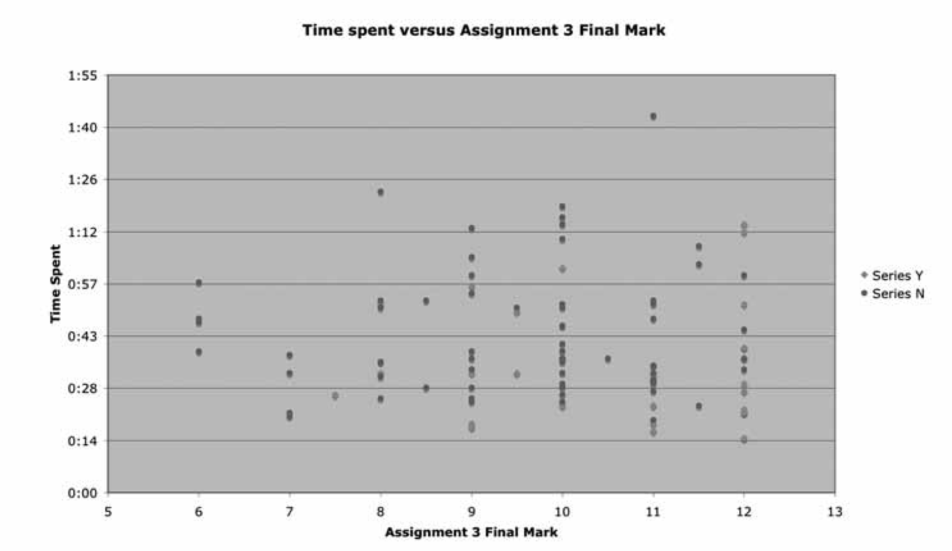

Figure 3: Final mark vs time spent (Cleaned dataset)

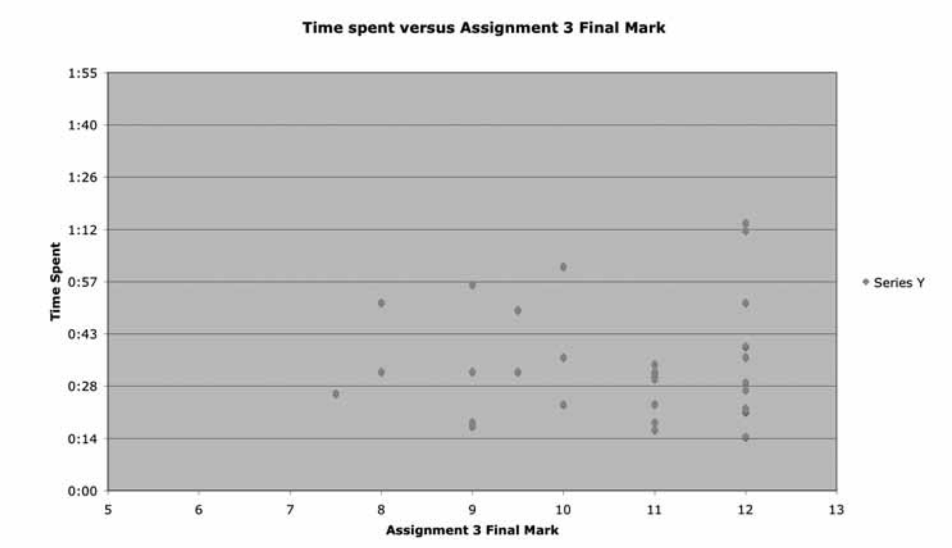

Figure 4: Final mark vs time spent (Y Only)

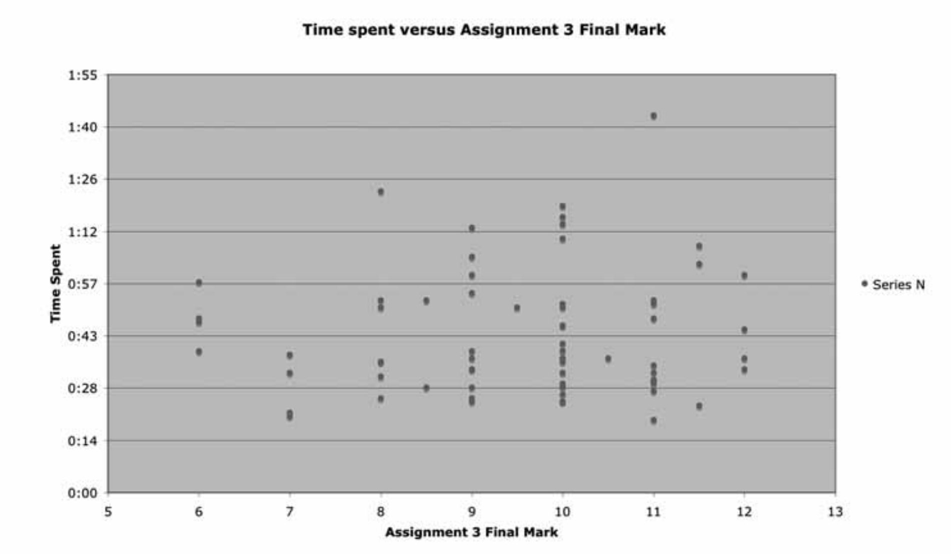

Figure 5: Final mark vs time spent (N Only)

The Rushing Student and the 11/12 Effect

In the 'Y Only' group, Figure 4, there is a concentration of marks in the range of 11-12 out of 12. The mean completion time for the 11/12 'Y Only' group was 27 minutes, compared to 35 minutes for the 12/12 group. To achieve a mark of 11, students had to achieve 3 Goods (4 out of 4) and 1 OK (3 out of 4) for the four parts. Reviewing the scripts revealed that the OK was almost always due to oversight, and the omission of a component, rather than error, where a component had been attempted incorrectly. Examining the students' CS1A examination marks revealed that these students, although they may achieve very high marks, did not achieve the same level of high marks as those students who received 12/12.

These results was compared with the 'N Only' group, Figure 5, who had far fewer 12/12 marks and a significant spike at 10/12. As the students who responded with 'No' were, supposedly, solving this problem for the first time, the time distribution was far more even for the results from 8 marks out of 12 to 12 marks out of 12, with 44 minutes the mean completion time. We hypothesise that this shorter time, and slightly lower mark, in the 'Y Only' group is due to the students rushing through the components of the practical and not taking the opportunity to review their work, or to read the question in detail and ensure that they answered all of the sub-parts. This appeared to be reflected in their CS1A examination results as well, potentially due to the same haste.

Overall Analysis

The distribution of final marks versus time sent may be fitted to a normal distribution in both cases but the 'Y Only' group have a significant skew to the left due to the large number of students who got it right. The 'N Only' group distributes around the 10/12 mark, with a fat left tail. This is reflected in the mean and standard deviation results obtained from analysis of the cleaned and separated data. Those students who believed that they had remembered the assignment were able to achieve a significantly higher score than those students who believed that they had to form the answer within the session.

Conclusions

We revisit our original hypotheses:

1. Students who remembered, or thought that they remembered, how they had solved the problem before would perform better in this assignment, due to improved comprehension.
2. Better students, as defined by their previous examination performance, would take less time to solve the problem due to their ability to apply their knowledge.
3. Students who perform better in examinations also tend to perform better in assignments.

There is clear and significant evidence, from Tables 1 and 2, to support our first hypothesis. Where a student has recalled the result they were more likely to have a higher mark and to complete the assignment in a shorter time. They have recall of the knowledge and they also have comprehension due to their ability to place that knowledge within the recognised framework of this assignment. An alternative hypothesis is that the students had memorised their solution from their initial attempt within the examination. While this is possible we consider it to be unlikely as the target question was placed at the end of the examination and the extracted component used for this assignment was a small part of the question, which was in turn approximately 20% of the examination. The placement and relative inconsequence of the question makes it an unlikely choice for memorisation, especially as there is no 'trick' to the question to reinforce a student's memory of their solution. Our experience in this area suggests that students need more than a few minutes, in a stressful situation, to memorise the solution to a question. We consider that the more likely hypothesis is that a student may have recalled the nature of the problem, rather than every fine detail, and that some synthesis was required to assemble the solution, as opposed to regurgitation. This conclusion is reinforced by the time taken for the student to write out the solution—had the solution been memorised a student should have been able to produce the solution in less than ten minutes, the time taken to write it on the paper. Both N and Y groups showed much mean larger times than this.

To assess hypothesis two, we must look at the students who claimed not to recall the solution. The mean CS1A mark does not vary significantly between the the different solution bands and there is also no clear distinction between the mean time taken to complete the assignment in each band. The largest group of high-performers in the 'N Only' group are found to have the answer 10/12 but they do not take a shorter time. There is no evidence to support our hypothesis that application skill, as measured by examination score, is correlated with the time taken to produce a solution. This is reinforced by the CS1A Mark versus Time Spent graph, Figure 1, that shows an accretion but no solid relationship between the two factors.

For hypothesis three, we have a clear relationship between CS1A score and final band, over all data, as shown in Figure 2. However, what we do not have is a reason for this correlation. Given that we have no relationship between time taken and mark for the 'N Only' group, we can eliminate time management as a significant contributor. We regard application as a factor due to the 'unseen' nature of this practical, although it was technically seen before under similar circumstances. There is, however, no correlation between CS1A mark and band result for the 'N Only' group. Therefore, the only distinction between groups is the one categorised by the student's recall of the previous solution and this problem – characterised as knowledge and comprehension. Students who knew the work and understood it in CS1A were significantly more likely to be able to repeat this act of recall and place it into the framework of this assignment (Bloom 1956). This familiarity also led to a reduction in the time taken to complete the assignment.

There has long been anecdotal evidence of a cadre of computer science students who are convinced of their own programming skills, an application level activity, but do not perform overly well in examinations where we assess their ability to express their comprehension and, in later subjects, build on their comprehension to assess analysis, synthesis and evaluation. This is not to say that they are bad programmers but their application is based firmly on their knowledge without the generalisation provided by a thorough comprehension. Another term for this form of application is 'boilerplating', where a known solution is fitted to a similar problem based on similarity of appearance rather than semantic similarity. By contrast, there are slow and methodical programmers who have excellent comprehension and a systematic, if unimaginative application, who can be relied upon to excel in examinations due to the knowledge and comprehension focus. Rather than resort to 'boilerplating', they will handmake new solutions. If students have truly moved through Bloom's taxonomy, then their application should be grounded in comprehension. The results of this experiment clearly illustrate that it is possible to provide evidence of application without necessarily reaching true comprehension. This is not surprising when the limited symbolic complexity and constraints implicit in programming are considered. This is one of the challenges that must be addressed when teaching students in computer science—their application must be grounded in true comprehension and assessment should test both levels of function.

Student Feedback

The anonymous feedback forms were used by students and comments were polarised between those students who saw the exercise as, effectively, a waste of their time and the majority of respondents who were positive about the experience. The few students who left the session early rarely made constructive comments and comments, when made, were angry in tone.

Overall, students were positive about working on inheritance and on receiving direct feedback from the demonstrators. Students noted that the assignment helped them to determine where their weaknesses lay and, while some students were quite abrupt about the waste of their time, the majority claimed that it was useful to them. Students who couldn't complete the question in the original examination were also positive about seeing a solution and having the opportunity for immediate feedback on their solution. Interestingly, from the time management perspective, some students claimed that, while it was interesting to see the question again, they only did badly in the exam because they ran out of time, not from any lack of knowledge.

Several students identified themselves as students who remembered their previous solution and often requested a more interesting, difficult or different practical. This is unsurprising as their recall and comprehension should have made this practical relatively trivial for them. Also, given the mark composition of students in the 'Y Only' group, these students are, in the majority, high achievers.

Future Work

Further refinement of this work is, unfortunately, difficult to undertake as the opportunity to test comprehension versus application only arose due to an unusual situation in an examination. However, further experiments are planned in the second half of this year with a combination of not-seen and previously-seen assignments to measure the impact of recall and comprehension versus application.

We are currently designing new assignments that cannot be solved with a purely mechanistic level of application but must employ Bloom's taxonomy and only allow application if comprehension has already been achieved. We are also developing assignments to assess the impact of haste and time management on students, to resolve the unanswered and unsupported hypotheses developed in this study.

We are also reviewing teaching approaches to determine what the correct target level should be for each course component and to align assessment with the development of knowledge, comprehension and then application.

The first year Puzzle-Based Learning course, to be introduced in 2009 by Professor Zbigniew Michalewicz, is designed to extend students by developing their ability to think about problem formation and solution selection. This occupies a higher cognitive domain, as it develops work at the evaluation level, when students are required to assess the suitability of one problem-solving technique over another. This will further develop the skills introduced in CS1B, which requires some analysis and a very template-driven evaluation approach, to provide students with alternate mechanisms for enhanced knowledge use and comprehension.

References

Biggs, J 1999, What the student does: teaching for enhanced learning, *Higher Education Research and Development*, 18(1), pp. 57–78.

Bloom, BS 1956, *Taxonomy of educational objectives: The classification of educational goals handbook*, Volume 1, McKay, New York.

Cannon, R & Newble, D 2006, *A handbook for teachers in universities and colleges*, 4th edn, RoutledgeFalmer.

Clear, T 2006, On the necessity of removing "cruelty" from the teaching of computing, *SIGCSE Bull.*, 38(2), pp. 8–10.

Dijkstra, EW 1989, The cruelty of really teaching computer science, *Communications of the ACM*, 32(12), pp. 1398–1404.

Dochy, F, de Rijdt, C & Dyck, W 2002, Cognitive prerequisites and learning: How far have we progressed since Bloom? Implications for educational practice and teaching, *Active Learning in Higher Education*, 3(3), pp. 265–284.

Ehmann, D 2005, Using assessment to engage graphic design students in their learning experience, in *Making a Difference: 2005 Evaluations and Assessment Conference Refereed Papers*.

Horstmann, C 2008, *Big Java*, 3rd Edition, Wiley.

Terenzini, PT, Theophilides, C & Lorang, WG 1984, 'Influences on students' perceptions of their academic skill development during college', *The Journal of Higher Education*, 55(5), pp. 621–636.

University of Cambridge 2007, *Computer Science Teaching Handbook*, Version 9, http://www.cl.cam.ac.uk/local/teaching-handbook.pdf.

Falkner, N 2009, Is comprehension or application the more important skill for first-year computer science students?, *ergo*, vol. 1, no. 2, pp. 15-26.

† Corresponding author: nickolas.falkner@adelaide.edu.au