Journal of Information Technology Education Volume 6, 2007

First Year Courses in IT: A Bloom Rating

Dave Oliver and Tony Dobele
School of Computing Sciences, Central Queensland University, Rockhampton, QLD, Australia

d.oliver@cqu.edu.au t.dobele@cqu.edu.au

Executive Summary

This paper explores the cognitive difficulty of assessment tasks in six first year computing courses within an Information Technology (IT) degree. This issue is pertinent to Information Technology education for two reasons. Degree level education in any field of study is expected to develop higher order thinking skills. Bloom's taxonomy is a framework which can be used to identify different levels of thinking skills. It calibrates ascending cognitive levels from the lowest, knowledge involving the recall of facts, to the highest, evaluation, which involves the comparative assessment of outcomes. Bloom's taxonomy can be used as a guide to designing a course of a requisite level. Bloom's taxonomy can also be used to calibrate the level of a particular assessment task retrospectively. Both approaches can be used for course planning and evaluation purposes, and, since a degree programme is simply a collection of courses, it is also possible to plan and measure at this level.

This paper follows the second tactic, as the analysis presented in the paper is based on the assessment components of the courses studied. This approach requires the researcher to self-reflect on the mental processes he or she would use to answer each question (or part question) in each item of assessment and to register the Bloom level of the processes used. Bloom levels range from 1 to 6 and the Bloom rating for a course is the weighted average of the Bloom levels assigned to each assessment question. We calculate a Bloom rating for each of the courses involved in the study.

This paper extends earlier work that examined the cognitive difficulty of two streams of three courses in an IT degree: a programming stream and a data communications and networking (DCN) stream. It was anticipated that within each stream there would be an increase in Bloom rating as the year level of a course advanced, since it was thought that later year courses would require higher level thinking skills than beginning ones. This expected gradient in Bloom ratings from early to later years was not confirmed by the analysis. The most striking feature of that analysis was a concentration of Bloom ratings for each stream and the wide difference between them. The Bloom rating for Programming courses were considerably higher than those in the DCN stream despite their placement at lower year levels in the degree.

This study takes a horizontal perspective of first year courses in an Information Technology degree. Retaining our previous assumption, that Bloom ratings of courses should increase as the year level increased, in this study, as all the courses were at the same year level, we expected the Bloom ratings of the courses to be similar. Again the findings

Editor: Janice Whatley

did not conform to expectations. This study showed more variation in Bloom rating in a single year than we anticipated, with ratings ranging from 1.3 to 4.8. However, when we looked at the pass rates, there was no indication that the courses with higher cognitive demands were harder to pass, since most of these courses had comparatively higher passing rates. We still maintain that courses should exhibit a rise in Bloom rating as they move downstream, since we expect the cognitive capabilities of students to improve as they advance through the degree programme. However, first year is important in laying the base from which more advanced work can be mounted. If first year courses have too high a cognitive level they may prevent students with lower ability levels from gaining a foundation from which to make upward progress.

We hope this study provokes reflection on the cognitive characteristics of courses and programmes with which readers are associated. We believe IT students should have well developed abilities at the application level and beyond by the time they graduate, but confidence in handling assessment tasks at the lower levels of Blooms taxonomy needs to be gained in first year before students are able to move onto assessment items that require the higher cognitive Levels of Analysis, Synthesis and Evaluation.

Keywords: Blooms Taxonomy, First year, IT, education, evaluation

Introduction

This paper builds on previous work by Oliver, Dobele, Greber and Roberts (2004a, 2004b) that examined the cognitive difficulty of two streams of three courses in an IT degree: a programming stream and a data communications and networking (DCN) stream. Both this study and the earlier study use Bloom's taxonomy as a measure of the cognitive difficulty of a course (Bloom, 1956), which is summarized in Table 2. Its purpose is to provide a classification of the cognitive depth required to perform a given task. It is very widely known and has been used as a reference point in a number of publications in IT including the IS2002 curriculum guidelines (Gorgone, Davis, Valacich, Topi, Feinstein, & Longenecker, 2002) and in academic papers (Box, 2004; Burgess, 2005; Howard, Carver, & Lane, 1996; Lister, 2001; Lister & Leaney 2003; Reynolds & Fox, 1996; Sanders & Mueller, 2000; Scott, 2003). The application of Bloom's Taxonomy adopted in these studies is outlined in more detail in the next section.

This study concerns the first year courses of an IT degree offered from our school. The presumption in our first study (Oliver et al., 2004a, 2004b), which we share with other writers as described in more detail subsequently, is that the Bloom rating of courses should progress from low ratings in the initial courses to higher levels in later year courses. Since this study is solely of first year courses, we expected that they would share similar Bloom ratings. As unfolded in our presentation of results later in the paper this expectation was not realized from our data. Possible reasons for why this was the case are explored at the end of the paper.

The structure of the paper is as follows. Firstly, we classify and describe a number of prior studies in IT that use Bloom's taxonomy. We then describe how we analyse the data and compute the Bloom rating for a course. Following this we present some sample data from the courses analysed and describe how it was classified according to Bloom's taxonomy. We then present the results of our analysis and discuss their implications. Finally, we make some concluding remarks on the outcomes of the study.

Applications of Bloom's Taxonomy

As mentioned in the introduction, Bloom's taxonomy is used as a reference point in a number of studies of Information Technology (IT) education. Table 1 shows applications of Bloom's taxonomy extend over a broad range of educational contexts from setting assessments to curriculum guidelines. The Association of Information Systems curriculum guidelines (Gorgone et al., 2002), uses a modified form of the taxonomy to describe the depth of treatment required for different topics in a degree programme. They propose “A topic may be covered at a low depth of knowledge level as part of an introductory course and in more depth (higher competency) in a subsequent course” (Gorgone et al., 2002). Reynolds and Fox, (1996) use Bloom’s Taxonomy to specify the cognitive difficulty of different educational objectives in subject areas within the ACM Curriculum ’91 guidelines. Their intent is to highlight the differing cognitive demands of different topics to ensure that those with higher cognitive demands are given adequate exposure in the curriculum, as they hold the belief that "objectives tend to concentrate at the lowest levels of mastery because they are the easiest ones to teach and test." Sanders and Mueller (2000) describe a curriculum design exercise for an entire degree based upon the principles of Bloom's taxonomy. Their proposal is consistent with that of Gorgone et al.(2002) in that they argue that courses in the early years of the programme should concentrate on achieving objectives set at the lower end of Bloom's taxonomy, whereas those in the final years should be oriented towards skills development at the upper end of the scale. This assumption underpins the work we have done. In Oliver et al.(2004a, 2004b) we examined the cognitive difficulty of two streams of three courses in an IT degree, a programming stream and a data communications and networking (DCN) stream using Bloom’s Taxonomy. That work identified a considerable discrepancy in the cognitive difficulty of the two streams. Also, the increased depth expected in later year courses that has been mentioned was not apparent in the DCN stream. The implications of Bloom’s taxonomy for a stream of courses is also followed by Lister (2001), who uses Bloom's taxonomy as a framework for formulating objectives for a sequence of programming courses. Lister and Leaney (2003) use Bloom’s Taxonomy to develop assessment items that test learning at different Bloom levels. Students may choose to complete only those assessment items necessary to gain a Pass, i.e. those items set at the knowledge and comprehension levels. To achieve a higher grade, students undertake assignments set at higher Bloom levels. This approach has also been followed by Box (2004) and Burgess (2005). Scott (2003) also uses Bloom’s Taxonomy to develop assessment but suggests that in any one test, questions should be drawn from all the Bloom levels, thus giving a better measure of student learning. Howard et al. (1996) use Bloom's taxonomy to evaluate the cognitive difficulty of the material in each lesson in a CS2 course.

Table 1. Application of Bloom’s Taxonomy in Computing Science

Application	Author
Curriculum Guidelines	(Gorgone et al., 2002; Reynolds & Fox, 1996)
Curriculum Development	(Sanders & Mueller, 2000)
Streams within a degree programme	(Lister, 2001; Lister & Leaney 2003; Oliver et al., 2004b, 2004a)
Assessments in a course	(Scott, 2003)
Lesson planning for a CS2 course	(Howard et al., 1996)
Determining grades in a course	(Box, 2004; Burgess, 2005; Lister & Leaney 2003)

Method of Analysis

The data used in both this and the earlier study are the summative assessment tasks used to determine the overall grade in a course, rather than the stated educational objectives. This approach requires the researcher to self-reflect on the mental processes he or she would use to answer each question (or part question) in each item of assessment and to register the Bloom level of the processes used. Bloom levels range from 1 to 6 and are shown in Table 2. As shown in Table 9, the number of questions that needed to be assessed ranged from 7 in the course with the fewest assessment items to 161 in the course that had the most assessment items. In all a total of 547 separate questions were assigned a Bloom level. The Bloom Rating for each course is the mean value obtained from multiplying the Bloom level of each question by the weighting allocated to it, as shown in the formula below. As two researchers analyzed the available data the results reflect the average of their individual analyses. The outcome from this process, discussed in detail subsequently, is shown in tables 4 to 6 and constitutes the main thrust of the paper.

$$\text{Bloom Rating} = \frac{\sum_{i=1}^{n} RiWi}{100}$$

where $\sum_{i=1}^{n} Wi = 100$, Ri is the Bloom Rating from 1 to 6 of assessment component i, Wi is the weight of component i, and n is the number of assessment components (Oliver et al., 2004a).

Table 2. Bloom's Taxonomy (Oliver et al., 2004a)

Level	Descriptor	Attainment level	Illustration of task from system modeling context
1	Knowledge	Ability to recall facts	Identify the symbols used in a specific modeling technique.
2	Comprehension	Understanding, Translation, Interpretation	Interpret a specification model. Identify inconsistencies or errors in specific models created using a specific technique.
3	Application	Use of knowledge in a new context	Create a specification model for a well-defined context.
4	Analysis	Identification of relationships	Create a specification model in a loosely defined but clearly delimited context.
5	Synthesis	(Re) Assembling of parts into a new whole	Create a specification model where the context and boundaries of a problem are undefined.
6	Evaluation	Making judgments	Determine the relative merits of alternative models. Assess whether a particular specification modeling technique is appropriate for a particular application.

The earlier analysis identified a clustering of Bloom Rating at the stream level. All the courses in the Programming stream generated an overall Bloom Rating between 3.3 and 4.0, a reasonably limited range and very distinct from the DCN stream values. The three DCN stream courses produced an overall Bloom Rating between 1.6 and 1.7, with a very small range of 0.1. The mean Bloom Rating for Programming was 3.7 whereas the mean rating for the Data Communications and Networking stream was 1.67, indicating a much higher cognitive requirement in assessment tasks for courses in the Programming stream (Oliver et al., 2004a, 2004b).

Courses Analyzed

This latest study explores the Bloom rating of six courses taught from within the School of Computing Sciences at Central Queensland University, which constitute three-quarters of the first year programme of study. Summaries of the content of these can be found in the Appendix. It revisits two of the courses which were in the original study, the two first year programming courses termed PROG1 and PROG2 in both studies, and a further four courses as shown in Table 3.

Table 3. Courses Analyzed

Course name	**Abbreviation**
Programming Fundamentals	PROG1
Procedural Programming	PROG2
Systems Analysis and Design	SAD
Software Fundamentals	SF
Conceptual Foundations of Computing	CFC
Workplace Issues	WI

A representative question drawn from each course is now presented with the aim of providing the reader with an appreciation of the style of assessment used in each course and also how each of the two researchers classified the question according to Bloom's taxonomy. The reader will note that in some cases the question was ranked differently by each researcher.

Programming Fundamentals (PROG1)

Which one of the following will NOT decrement the variable num by one?

```
(A)   num = num - 2 + 1 ;
(B)   --num;
(C)   num += num + 1 - num ;
(D)   num += -1 ;
```

To answer this question the respondent needs to be able to identify the variable *num* in each of the following statements and note the effect each statement has on the value in *num.* It is a little more advanced than the knowledge classification would be in our view, but not by a great deal. The respondent needs to work out the effect of each statement in order to identify the correct answer. Both researches rated this question 2 (comprehension). The question is not sufficiently advanced to be coded application since the context is completely defined by the question.

Procedural Programming (PROG2)

An ASCII text file, items.dat, contains a list of items in stock. The file has the name of a stock item on one line followed on the next line in the file by the number of that item in stock and the price of that stock item. There may be any number of items in the file and the file is terminated by an EOF marker. You are to develop a C++ program to read these items from the text file and display them

If the text file contains the data,

Sand paper, 7,1.25
Chain saw, 3,455.99
Hammer drill, 10, 25.99

then your program should display:
Items in stock

```
Sand paper   -  7 $1.25
Chain saw    -  3 $455 .99
Hammer drill - 10 $25.99
```

Your display should not be formatted.

This question was coded 3 (application) by researcher 2 and 4 (analysis) by researcher 1 making an average of 3.5. This question requires the student to comprehend a simple case study and apply concepts of C++ programming to code a program to solve the problem described. Researcher 1 felt that analysis of the problem was necessary in order to develop an algorithm to solve it and consequently rated the question at the analysis level of Bloom's taxonomy.

Systems Analysis and Design (SAD)

Technologies that enable the breaking of long held business rules that inhibit organisations from making radical business changes best defines:

a. Joint Application Design.
b. Rapid Application Development.
c. structured programming.
d. business process reengineering.
e. disruptive technologies.

This question simply asks respondents to recall previously learned material so was coded 1 (knowledge) by both researchers.

Software Fundamentals (SF)

When using odd parity to detect communication errors, the total number of 1s in a bit pattern (including the parity bit) must be odd.

This question is of the True/False type. Again, in order to answer it this question asks respondents to recall previously learned material so was coded 1 (knowledge) by both researchers.

Conceptual Foundations of Computing (CFC)

Simplifying the expression $\frac{\left(2x^2y\right)^4}{8x^5y^{-4}}$ **(where** $x \neq 0$**) and leaving your answer with positive exponents gives**

A) $2x^3y^8$

B) $2x^3$

C) $\frac{y^5}{4x}$

D) $\frac{x}{4y^3}$

This question requires students to have learnt how to manipulate exponents, then to use that knowledge in the same context to simplify an expression. Both researchers rated the question at level 2 (comprehension) since students' knowledge was not required to be applied in a new situation or in a different way.

Workplace Issues (WI)

You work for a company that has recently designed and developed a computerised medical device. This device hooks directly into patients suffering from a particular disease. At regular intervals, the device will monitor the patient and automatically inject required drug dosages into the relevant patient. Clinical trials have shown that 60% of patients using these devices have completely recovered from the disease. However, the other 40% have shown symptoms that are worse than before. Even though unproven, it is highly suspected that the worsening condition of these remaining 40% of patients is also a direct result of using the device. Should you or should you not make a recommendation to your company to release this device to the public? Discuss your case.

This course was the only one of the six studied where any assessment items rated higher than 5 (synthesis). This question was coded 6 (evaluation) by both researchers since it required the respondent to make a judgment given certain information. In addition it may reasonably be assumed that "discuss your case" requires the respondent to defend their position using knowledge and arguments acquired during the course, which is another requirement of tasks at the evaluation level.

Results

As shown in Tables 4 and 5 the breakdown between assignments and examinations is very consistent in the courses studied. The data in Table 4 indicates SF has the lowest Bloom rating for assignments and WI the highest. SF has the highest percentage of knowledge level assignments and WI the lowest. CFC has most assessment at the Comprehension level. SAD and PROG1 have a fairly even balance of assessments at levels 1, 2 and 3. The preponderance of assessments at levels 1-3 for all of the courses seems appropriate to us for first year courses. WI records a high Bloom rating compared to the other courses and exceeds our expectations for a first year course.

PROG1 and PROG2 register lower Bloom ratings for assessment here than in the earlier study where they were 3.9 and 4.5 respectively. In our assessment this is a positive move in helping students ease into the complex area of computer programming. SAD, PROG1 and PROG2 also have some assessment at level 3, application, which we consider a desirable level of attainment for IT courses due to the applied nature of the discipline.

Table 4. Bloom Rating: Assignments

Course	Knowl-edge	Compre-hension	Applica tion	Analysis	Synthesis	Evalua-tion	Weight-ing	Bloom Rating
SF	30.5	5.3	0.2	0.0	0.0	0.0	36.0	1.2
CFC	7.3	32.6	0.0	0.1	0.0	0.0	40.0	1.8
SAD	10.0	8.5	15.3	2.0	4.2	0.0	40.0	2.6
PROG1	8.8	9.9	11.3	10.0	0.0	0.0	40.0	2.6
PROG2	3.8	4.7	15.0	11.5	0.0	0.0	35.0	3.0
WI	1.5	3.0	0.0	19.5	6.0	10.0	40.0	4.4

Table 5. Bloom Rating: Examinations

Course	Knowl-edge	Compre-hension	Applica tion	Analysis	Synthesis	Evalua-tion	Weight-ing	Bloom Rating
SF	42.0	22.0	0.0	0.0	0.0	0.0	64.0	1.3
SAD	42.0	15.0	3.0	0.0	0.0	0.0	60.0	1.4
CFC	12.0	48.0	0.0	0.0	0.0	0.0	60.0	1.8
PROG1	13.2	22.8	12.0	12.0	0.0	0.0	60.0	2.4
PROG2	4.0	27.0	15.5	18.5	0.0	0.0	65.0	2.8
WI	0.0	6.0	0.0	18.0	0.0	36.0	60.0	5.0

Table 6. Bloom Rating: Overall

Course	Knowl-edge	Compre-hension	Applica tion	Analysis	Synthesis	Evalua-tion	Weight-ing	Bloom Rating
SF	72.5	27.3	0.2	0.0	0.0	0.0	100.0	1.3
SAD	52.0	23.5	18.3	2.0	4.2	0.0	100.0	1.8
CFC	19.4	80.4	0.0	0.2	0.0	0.0	100.0	1.8
PROG1	22.0	32.7	23.3	22.0	0.0	0.0	100.0	2.5
PROG2	7.8	31.7	30.5	30.0	0.0	0.0	100.0	2.8
WI	1.5	9.0	0.0	37.5	6.0	46.0	100.0	4.8

It seems reasonable to expect a lower level of difficulty for examinations due to the limited time available to the student in this form of assessment. Comparing tables 4 and 5, the Bloom rating for examinations is slightly higher in SF, but only marginally, and at 1.3 is the lowest Bloom rating of all of these courses. For SAD there is a reduction from 2.6 to 1.4, for PROG1 there is a reduction from 2.6 to 2.4 while CFC remains the same at 1.8. PROG2 drops from 3.0 to 2.8 and only WI goes up from 4.4 to 5.0.

Continuing to compare tables 4 and 5, we can see that the proportion of level 1 assessment in SF declines from 84.7% (30.5/36 *100) to 65.6% (42/64*100) and the proportion of level 2 assessment goes up from 14.7% to 34.4% but the Bloom ratings are low compared to the other courses. The proportion of assessment at level 4 in PROG1 drops from 25.0% in assignments to 20.0% in the examination and in PROG2 from 32.9% to 28.5%, and the proportion of assessment at level 3 in PROG1 drops from 28.3% in assignments to 20% in the examination and in PROG2 from 42.9% to 23.8% which is more in keeping with the anticipated trend of lower cognitive requirements required in examinations.

Looking at the overall Bloom ratings in Table 6, SF, with the lowest Bloom rating of 1.3, has most assessment (72.5%) at level 1 and 27.3% at level 2. SAD and CFC are next on 1.8 overall but whereas CFC has 80% of assessment at level 2, which is very close to its overall Bloom level, SAD has assessments spread more broadly but mostly over the lower three levels. PROG1 and PROG2 are spread more widely still over levels 1, 2, 3 and 4, which lead to the higher overall Bloom ratings.

In comparing the results of this study and the earlier one it is interesting to note a significant change in the Bloom ratings of PROG1 and PROG2. In the earlier study their Bloom rating was 3.9 and 4.0 respectively and the suggestion made in that analysis was that these ratings were on the high side. The current ratings of 2.5 and 2.8 obtained here for the two equivalent courses in Term 2, 2006 indicates a softening of these courses has occurred to a level we consider more appropriate for first year. We are not able to conclude that this reduction in the Bloom level has an impact on the passing rate for these courses. As indicated in Table 7, the pass rate for PROG1 in T2 of 2006 is slightly lower than it was in T2 of 2002. However the pass rate for PROG2 is much higher, increasing from 45.7 to 69.8.

In addition as shown in Table 8, the Bloom rating of a course and its success rate do not appear to correlate well. WI which has the highest Bloom rating has the third highest passing rate and SF which has the lowest Bloom rating has the lowest passing rate. The scatter diagram shown in Figure 1, achieved from plotting the Bloom Rating and the pass rate from the six courses in this study and the six in the previous study, is similarly inconclusive. The correlation coefficient *r* of -0.25 indicates that, for our research so far, there is no apparent relationship between the Bloom rating and the student pass rate.

WI is something of an outlier. The aims and objectives incline more to the social than the technical, which differentiates it from the other courses, but why this should lead to such a high Bloom rating is not clear. It is perhaps not a typical first year IT course. It has been developed from an earlier course called Professional Issues which was scheduled in the third year, and has possibly inherited characteristics of third year courses which are supposed to have greater depth. However this is purely a pragmatic explanation.

The profile of the assessment tasks in the six courses is varied. As shown in Table 9, CFC with 161, SF with 133, and PROG1 with 129, are characterized by a large number of assessable items. WI has the least at 7 followed by PROG2 with 43 and SAD in the middle with 74. There is some indication that fewer, and hence more expansive questions, are associated with greater cognitive requirements as WI has the fewest assessable items and the highest Bloom rating.

Table 7. Pass Rate for PROG1 & PROG2		
	Passing Grades%	**Non-Passing Grades%**
PROG1, T2, 2002	66.5	33.5
PROG1, T2, 2006	60.1	39.9
PROG2, T1, 2003	45.7	54.3
PROG2, T2, 2006	69.8	30.2

Table 8. Results T2, 2006		
	Passing Grades%	**Non-Passing Grades%**
SAD	76.0	24.0
PROG2	69.8	30.2
WI	68.9	31.1
PROG1	60.1	39.9
CFC	55.2	44.8
SF	51.9	48.1

Table 9. Assessable items														
	PROG1					**PROG2**					**SAD**		
	A1	**A2**	**A3**	**Exam**		**A1**	**A2**	**A3**	**Exam**		**A1**	**A2**	**Exam**
Assessment %	15	10	15	60		5	15	15	65		20	20	60
Nr. of items	100	3	2	24		10	1	1	31		18	17	39
Total items	129					43					74		
	SF					**CFC**					**WI**		
	A1	A2	A3	Exam		A1	A2	A3	Exam		A1	A2	Exam
Assessment %	12	12	12	64		15	15	10	60		15	25	60
Nr. of items	24	24	24	61		54	40	30	37		2	2	3
Total items	133					161					7		

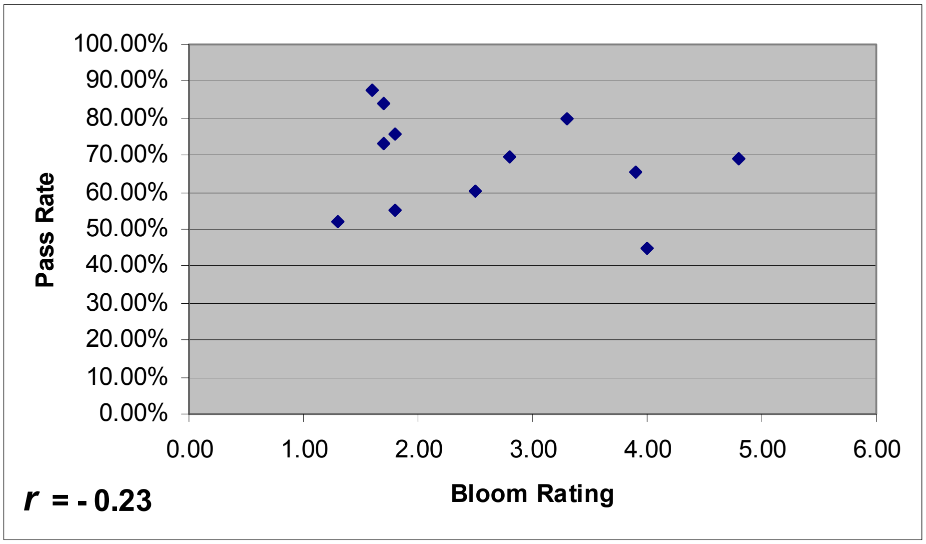


Figure 1: Bloom Rating to Pass Rate

Conclusion

In the context of the degree programme in which they are situated, PROG1, PROG2, SF and SAD can very clearly be seen to lead into later year courses. This could mean their Bloom levels are influenced by their year placement. CFC and WI provide a more general base to the degree as a whole and do not have such a close linkage to later year courses. It is possible this relative detachment allows CFC and WI to have more freedom to follow the assessment styles typical to their field of study, which in turn tends towards certain Bloom ratings.

This study has extended the application of the Bloom rating into courses beyond programming and data communications and networking. Further, this study has explored the Bloom rating of first year courses, which could be expected to occupy similar Bloom levels. The previous study compared streams, where an increase in Bloom rating was anticipated as the year level of a course rose. In both cases the results confounded expectations. In the earlier study the expected gradient in Bloom ratings from early to later years was not observed and this study shows more variation in Bloom rating in a single year than was expected. We still maintain that courses should exhibit a rise in Bloom rating as they move downstream, since we expect the cognitive capabilities of students to improve as they advance through the degree programme. However, first year is important in laying the base from which this advance can be mounted. If first year courses have too high a cognitive level they may prevent students with lower ability levels from gaining a foundation from which to make upward progress.

We hope this paper stimulates readers to further reflect on the cognitive requirements of assessments items they set. IT students should have well developed abilities at the application level and beyond by the time they graduate, but usually competence in handling assessment tasks at the lower levels of Blooms taxonomy needs to be gained in first year before students are able to move onto assessment items that require the higher cognitive levels of Analysis, Synthesis and Evaluation.

Acknowledgements

Special thanks to those colleagues in the School who assisted us in providing information relating to the courses used in this study.

References

Bloom, B. S. (Ed.). (1956). *Taxonomy of educational objectives, Handbook 1: Cognitive domain*. New York: Longman.

Box, I. (2004). Object-oriented analysis, criterion referencing, and Bloom. Paper presented at the *Sixth Australasian Computing Education Conference*, Dunedin, New Zealand.

Burgess, G. A. (2005). Introduction to programming: Blooming in America. *Journal of Computing Sciences in Colleges, 21*(1), 19-28.

CQU. (2006a). Conceptual Foundations of Computing. Retrieved 24/11/2006, from http://webfuse.cqu.edu.au/Courses/2006/T2/COIT11118/

CQU. (2006b). Procedural Programming. Retrieved 24/11/2006, from http://webfuse.cqu.edu.au/Courses/2006/T2/COIT11134/

CQU. (2006c). Programming Fundamentals. Retrieved 24/11/2006, from http://webfuse.cqu.edu.au/Courses/2006/T2/COIT11133/

CQU. (2006d). Software Fundamentals. Retrieved 24/11/2006, from http://webfuse.cqu.edu.au/Courses/2006/T2/COIT11221/

CQU. (2006e). Systems Analysis and Design. Retrieved 24/11/2006, from http://webfuse.cqu.edu.au/Courses/2006/T2/COIT11166/

CQU. (2006f). Workplace Issues. Retrieved 24/11/2006, from http://webfuse.cqu.edu.au/Courses/2006/T2/COIT11223/

Gorgone, J. T., Davis, G. B., Valacich, J. S., Topi, H., Feinstein, D. L., & Longenecker, H. E. (2002). *IS2002 model curriculum and guidelines for undergraduate degree programs in information systems.* Association of Information Systems (AIS).

Howard, R. A., Carver, C. A., & Lane, W. D. (1996). Felder's Learning Styles, Bloom's Taxonomy, and the Kolb Learning Cycle: Tying it all together in the CS2 course. *ACM SIGCSE Bulletin, 28*(1), 227-231.

Lister, R. (2001). Objectives and objective assessment in CS1. *ACM SIGCSE Bulletin, 33*(1), 292-296.

Lister, R., & Leaney, J. (2003). Introductory programming, criterion-referencing and Bloom. *ACM SIGCSE Bulletin, 35*(1), 143-147.

Oliver, D., Dobele, T., Greber, M., & Roberts, T. (2004a). Comparing course assessments: When lower is higher and higher, lower. *Computer Science Education, 14*(4), 321-341.

Oliver, D., Dobele, T., Greber, M., & Roberts, T. (2004b). This course has a Bloom rating of 3.9. Paper presented at the *Sixth Australasian Computing Education Conference*, Dunedin, New Zealand.

Reynolds, C., & Fox, C. (1996). Requirements for a computer science curriculum emphasising information technology subject area: Curriculum issues. *ACM SIGCSE Bulletin, 28*(1).

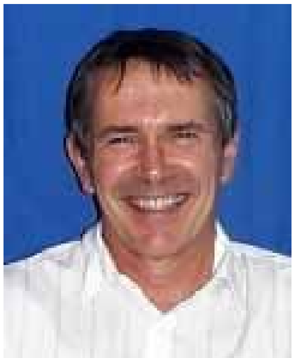
Sanders, I., & Mueller, C. (2000). A fundamentals-based curriculum for first year computer science. *ACM SIGCSE Bulletin, 32*(1).

Scott, T. (2003). Bloom's taxonomy applied to testing in computer science classes. *Journal of Computing Sciences in Colleges, 19*(1), 267-274.

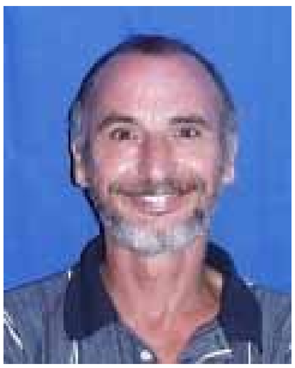
Appendix

Course	Synopsis
PROG1	This course is designed to teach commencing students structured programming principles. The course introduces the student to computer programming and the use of C++ programming language as a tool to implement structured programs from supplied software designs (CQU, 2006c)
PROG2	This course aims to introduce the student to more advanced computer programming using the C++ language. As well as learning C++, the student will learn how problems that are initially described in very general terms can be analysed, outlined and finally transformed into well-organised C++ programs using structured programming techniques. The course also introduces students to object-oriented programming in C++. The assessment items are designed to reward regular study throughout term and to provide timely feedback to students. (CQU, 2006b)
SAD	This course develops knowledge of basis systems analysis and design techniques including feasibility studies, fact-finding techniques, data and process modelling, and broad systems design. A range of development methods is presented, including both structured and object-oriented approaches and an introduction to UML. It includes the use of appropriate modelling techniques and software tools. The role of CASE tools in the analysis and design process will also be addressed. The application of these methods to current systems such as Web Information Systems is considered. (CQU, 2006e)
SF	The objective of the course is to provide students with a broad understanding of computer software: the differences between application software and system software; how system software controls the computer hardware; how system software simplifies use of the computer hardware for application programmers; how system software controls use of the computer hardware by multiple applications; how systems software prevents one application from interfering with another application; how data is stored and managed in a computer; and much more. This course will help students to develop an understanding of the relevance of skills like programming in the IT discipline. Specifically, it will provide an understanding of the software environment that hosts the programs that students will develop in first year programming courses, and the applications and systems that students will study in other first year IT courses. (CQU, 2006d)
CFC	The purpose of this course is to introduce some of the mathematics underlying the operation of computer systems and provide a foundation in elementary computer mathematics topics. As well as a review of basic mathematics, the course covers a number of discrete mathematics topics including number systems and arithmetic, data representation, algebra, sets, propositions and logical operators, boolean circuits, graphs, sequences, combinations and permutations. (CQU, 2006a)
WI	The aim of this course is to prepare students for a role as an informed professional within the IT/IS industry. It addresses the legal, social and ethical issues relating to the evolution of computer technology within society. It also introduces the student to a career in IT and what is involved in the various disciplines with the IT industry. (CQU, 2006f)

Biographies



Dave Oliver is a Senior Lecturer in Computing in the Faculty of Business and Informatics, Central Queensland University, Queensland, Australia. He has an Honours degree in Economics from Warwick University, UK; a Master of Technology degree in Computer Science from Brunel University, UK; and a doctoral degree from Central Queensland University. Most of Dave's publications are in the following areas: ERP Adoption, Computer Science Education, e-Commerce and Virtual Teamworking. Dave has also worked as a computer programmer and lectured in the UK.



Tony Dobele is a Lecturer with Central Queensland University, Faculty of Business and Informatics, and is based on the Rockhampton campus. He has a Bachelor of Information Technology and a Master of Information Systems and lectures in those areas. Tony's interests include data mining, machine intelligence and programming. He has a background in the finance and retail industries.