This Course Has A Bloom Rating Of 3.9

Dave Oliver, Tony Dobele, Myles Greber, Tim Roberts

Faculty of Informatics and Communication, Central Queensland University, Australia, 4702

`{d.oliver,t.dobele,m.greber,t.roberts}@cqu.edu.au`

Abstract

This paper analyses the cognitive difficulty of six courses that may be taken as credit towards an IT degree offered by a regional Australian University. The assessment requirements of these six courses are evaluated using Bloom's taxonomy and from this a difficulty metric, called here a *Bloom Rating*, is computed for each course. The analysis reveals that some quite lowly courses in terms of their ordering in the programme, such as first-year programming, are comparatively high in their cognitive demands, whereas some of the more advanced non-programming courses have relatively low levels of cognitive difficulty. An explanation for these trends is offered.

Keywords: Blooms Taxonomy, Programming, CS1, Data Communications, Networks

1 Introduction

Bloom's taxonomy is a mature technique for analysing the cognitive depth of performing a given task. It is self-described as a "concise model for the analysis of educational outcomes in the cognitive area of remembering, thinking, and problem solving." (Bloom, 1956 p2). It was specifically developed for use in educational contexts and has been used as a basis for a number of studies of IT education. Howard et al (1996) use Bloom's taxonomy to perform a per lesson depth analysis of a CS2 course. Sanders and Mueller (2000) describe a curriculum design exercise for an entire degree based upon the principles of Bloom's taxonomy. 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 orientated towards skill development at the upper end of the scale. Lister (2001), writing from the perspective of a CS1 lecturer, follows this theme and uses Bloom's taxonomy as a framework for formulating course objectives for a sequence of programming courses as summarised in Table 1. This merging of objectives as favoured by Lister has also been performed in other works based on Bloom's taxonomy including the Association of Information Systems (AIS), (Gorgone et al., 2002). This tendency to merge objectives may partly be a response to the difficulty acknowledged by Bloom "in finding a method of classification which would permit complete and sharp distinctions among behaviors" (p15). For the purposes of this study the original classification is used.

Year	Level	Objectives
1	1 and 2	reading and comprehending code
2	3 and 4	writing code fragments for a defined context
3	5 and 6	writing complete non-trivial programs

Table 1. A Programming stream over 3 years structured according to Bloom's taxonomy

The faculty of the university at which this study was undertaken does not have a strong appreciation or commitment to the application of Bloom's taxonomy for degree design. This is not to say that individual lecturers, especially those who have graduated from teacher education programmes, are not aware of it and employ these principles in courses they design.

Bloom's taxonomy is shown in Table 2. It suggests tasks may be ranked according to their conceptual difficulty (as previously stated, the general implications for educators are that lower level learning objectives and assessment tasks are more suitable for beginning students whereas the higher-level objectives are more suited to upper level courses). However attainment of the higher levels suggests the lower levels must be traversed first. It is important in a degree programme in Information Technology to foster abilities at the application level and to develop critical thinking skills, which are most evident at the higher levels of Bloom's taxonomy. One would expect these characteristics to be relatively more pronounced in upper as opposed to lower level courses (as defined by level within the degree programme).

The speed of transition that is feasible from low to high-level objectives is likely to be conditioned by the perceived abilities of the student body at any particular institution. Elite institutions that admit only students with very high academic attainments are more likely to attempt swifter transitions to higher levels than those with a broad based entry.

Level	Descriptor	Attainment level
1	Knowledge	Ability to recall facts
2	Comprehension	Understanding, Translation, Interpretation
3	Application	Use of knowledge in a new context
4	Analysis	Identification of relationships
5	Synthesis	(Re)Assembling of parts into a new whole
6	Evaluation	Making judgements

Table 2. Bloom's Taxonomy

Bloom indicates that "objectives may also be inferred from the tasks, problems, and observations used to test or evaluate…" (1956 p12). This is the approach taken in this study where educational objectives in terms of their Bloom classification are derived from assessment tasks in a range of courses. Bloom has used this approach in his manuscript by clarifying "the behavior appropriate to each category by illustrations of the examination questions and problems which are regarded as appropriate." (1956 p44).

2 The courses that were studied

Six courses from a single university were selected for this analysis. Three of these courses constitute a C++ programming stream (PROG1, PROG2 and PROG3) and three constitute a Data Communications and Networking stream (DCN1, DCN2 and DCN3). It should be noted here that the term *stream* is used here as opposed to major or minor, since the degree within which these courses are hosted has a very open structure. There is no need for a student to complete all courses within a stream, although the first course in each of these streams is a compulsory requirement for the degree as a whole. The programming stream has a linear structure, as there is a strict order in which each course in this stream must be studied. In contrast the Data Communications and Networking stream is tree structured, suggesting the stream provides breadth rather than depth.

Their relationship to each other and the designated level of these courses within the degree as a whole is shown in Table 3. It is important to note here that the levels in Table 3 are not the levels that Bloom defines. These levels are associated with the six levels into which a three-year programme with two semesters per year is structured as shown in Table 4.

Level	Courses in the C++ Programming Stream	Courses in the Data Communications and Networking Stream
1	PROG1	
2	PROG2	
3	PROG3	DCN1
4		DCN2, DCN3

Table 3. Stream Structure

These levels have an implicit meaning to lecturers involved in the programme. There is an informal reckoning as to what is appropriate in terms of assessment tasks at each level; however the courses are not explicitly ranked according to Bloom's or any other taxonomy.

Level	Year	Semester
1	1	1
2	1	2
3	2	1
4	2	2
5	3	1
6	3	2

Table 4. Relationship between year and level

The course offerings used for the study were from the first half of 2003 with the exception of PROG1, which was the second half of 2002. Four researchers from the university at which these courses are offered took part in the study. Each researcher was a lecturer who had some experience of at least one of the courses that were studied, though not necessarily for the specific course offering that was selected. It is perhaps useful to note that it is quite common for lecturers in the faculty to teach courses where another lecturer has devised the assessment. The main advantage derived from using a number of researchers is that the charge of subjectivity that could be levelled if a single researcher performed the analysis is negated. This is important in an analysis of this type, which requires academic judgement in the categorisation of assessment items according to the taxonomy.

In this study, for each of the courses selected, each question in each assessment item (both assignments and scheduled tests and examinations) was classified from 1 to 6 on Bloom's scale by each researcher independently. Where a question had parts, the analysis was applied to each part separately. Each question (or part of question) constitutes a certain weighting of the overall assessment. A weighted average of the consolidated rankings of all of the four researchers for each course was then calculated giving a Bloom rating for each course. This may be described by the following formula.

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

Where R is the Bloom classification from 1 to 6 of an assessment component W is the weight of the component and n is the number of assessment components in a course.

The most notable feature of the way the analysis was approached was that one of the researchers was much more discriminating in the classification of questions and would allocate a range of cognitive levels to a given task. For example consider PROG2 exam question 2, which asked the output from the following C++ code.

```
int s = 17, m = 101, k;
k = ((s | m) >> 2);
cout << k;
```

Table 5 shows how the four researchers coded PROG2 exam question 2. It can be seen that whereas researchers 1, 3 and 4 assigned a single Bloom classification to this question, researcher 2 adjudged the question to require knowledge, comprehension and application skills in the proportions 2/5, 2/5 and 1/5 respectively.

3 Results

3.1 Introduction

The Bloom rating for a course is the value shown in the rightmost column in Tables 6-8. All entries in this column will be a number between one and six and indicate the overall Bloom rating for each course. Column one of Tables 6-8 lists each of the six courses PROG1, PROG2, PROG3, DCN1, DCN2 and DCN3 respectively. Columns two to seven show the percentage assessment content of each course in the six categories of Bloom's taxonomy, Knowledge, Comprehension, Application, Analysis, Synthesis and Evaluation. Table 6 shows the results for assignments, Table 7 for examinations and Table 8 shows the overall Bloom rating for each course. Each cell shows the percentage assessment; for example in Table 6 the Application column, PROG3 has an entry of seven. This indicates that, considering assignments only, seven percent of the overall assessment in PROG3 was at the Application level of Bloom's taxonomy. Column eight shows the overall weighting attached to this assessment type, which should be equal to the sum of columns two to seven.

3.2 Assignments

Table 6 shows the results of the analysis for assignments completed in the student's own time. These results are typical for the study as a whole in that they show a clustering of cognitive activity at the knowledge and comprehension levels for the DCN stream whereas the cognitive demands in the Programming stream are more diverse and have a much stronger presence in the application, analysis and synthesis levels.

3.3 Examinations

Table 7 shows the results of the analysis for examinations. PROG1 does not have any assessment by examination so no results appear in that row. It is interesting that the Bloom rating for PROG3 for examinations is higher than for assignments as examinations may be regarded as intrinsically more demanding, as they are constrained with respect to time and place. PROG2 places comparatively lower cognitive demands in examinations, which is perhaps more appropriate. The DCN stream also presents a mixed set of comparisons. DCN1 and DCN3 also make greater cognitive demands in examinations whereas DCN2 does not. The DCN stream shows a preponderance of assessment at the knowledge and comprehension levels.

3.4 Overall comparisons

A number of observations can be made from this analysis. Firstly, in Table 8 clustering of Bloom rating is clearly discernable at the stream level. All the programming courses have an overall Bloom rating between 3.3 and 4.0, which is quite a limited range and totally distinct from the Data Communications and Networking stream values. The three Data Communications and Networking stream courses have an overall Bloom rating between 1.6 and 1.7, exhibiting a remarkably small range of 0.1. The rating for each stream differs from the other to a much greater degree than that between courses within a stream. Also the mean Bloom rating for Programming is 3.7 whereas the mean rating for the Data Communications and Networking stream is 1.67, indicating a much higher cognitive requirement in assessment tasks for Programming. This suggests that the cognitive levels required to accomplish tasks within these two streams of IT are very different. It also qualifies the assumption made by Reynolds (1996) that "objectives tend to concentrate at the lowest levels of mastery because they are the easiest ones to teach and test" (p249). It appears that in the programming stream, low-level objectives are more difficult to teach and test than high-level objectives. Possibly this is due to the fact that most course designers have considerable expertise in programming, and as a consequence find it difficult to conceive of low-level learning objectives. On the other hand we must consider the cognitive demands being placed on learners. Conversely assessment tasks at the application level and above in Data Communications and Networking are relatively limited. For DCN1, DCN2 and DCN3 respectively 11.9%, 13.8% and 13.1% of the assessment is at this level. A very small proportion (less than 1% in each case) is at levels 5 and 6, however these proportions are appropriately located in the upper level courses DCN2 and DCN3. These results may be due to the extensive factual base of these courses and the conventions in place for assessing this discipline area. In a sense this is a

Prog2 Exam Q2	Knowledge	Comprehension	Application	Analysis	Synthesis	Evaluation
Researcher1		1				
Researcher2	0.4	0.4	0.2			
Researcher3				1		
Researcher4			1			

Table 5 Coding of PROG2 Exam Q2

Course	Knowl-edge	Compre-hension	Applica-tion	Analysis	Synthesis	Evalua-tion	Weight-ing	Bloom Rating
PROG1	6.9	6.6	29.3	7.6	49.6	0.0	100%	3.9
PROG2	2.8	1.5	5.9	4.3	20.5	0.0	35%	4.5
PROG3	7.8	14.9	7.0	0.6	9.8	0.0	40%	2.7
DCN1	19.7	11.7	0.5	3.2	0.0	0.0	35%	1.6
DCN2	16.9	13.5	3.3	1.0	0.0	0.3	35%	1.9
DCN3	32.3	6.5	0.9	0.3	0.0	0.0	40%	1.2

Table 6. Bloom Rating: Assignments

Course	Knowl-edge	Compre-hension	Applica-tion	Analysis	Synthesis	Evalua-tion	Weight-ing	Bloom Rating
PROG1	N/A	N/A	N/A	N/A	N/A	N/A	0%	N/A
PROG2	5.1	5.8	12.3	6.8	31.1	2.0	65%	3.9
PROG3	5.3	9.7	13.2	5.3	25.0	1.5	60%	3.7
DCN1	30.8	25.9	0.9	7.4	0.0	0.0	65%	1.8
DCN2	33.3	22.5	5.1	3.6	0.6	0.0	65%	1.7
DCN3	28.1	20.1	8.9	2.6	0.2	0.2	60%	1.8

Table 7. Bloom Rating: Examinations

Course	Knowl-edge	Compre-hension	Applica-tion	Analysis	Synthesis	Evalua-tion	Weight-ing	Bloom Rating
PROG1	6.9	6.6	29.3	7.6	49.6	0.0	100%	3.9
PROG2	7.8	7.3	18.2	11.1	51.6	2.0	100%	4.0
PROG3	13.0	24.6	20.2	5.9	34.8	1.5	100%	3.3
DCN1	50.5	37.6	1.3	10.6	0.0	0.0	100%	1.7
DCN2	50.1	36.0	8.4	4.5	0.6	0.3	100%	1.7
DCN3	60.4	26.6	9.8	2.9	0.2	0.2	100%	1.6

Table 8. Bloom Rating: Overall

disappointing discovery as Bloom himself writes that: "The fact that most of what we learn is intended for application to problem situations in real life is indicative of the importance of application objectives in the general curriculum" (1956 p122).

4 Concluding Remarks

This is a study made at a single institution over a relatively small group of courses. A cross-institutional study to determine whether the data presented here is anomalous would be instructive. Studies could also be made of other streams within IT such as Database, Electronic Commerce, Multimedia and so on to see what Bloom characteristics they exhibit.

As far as this analysis is concerned there seem to be some anomalies in the way courses are categorised as advanced or not advanced. The Bloom analysis would practically reverse the conventional ordering. For if we invert the rows of PROG1 and PROG2 in Table 8 so the table is in descending order of Bloom rating, it is almost in the same sequence as Table 3, although you would expect, from a perusal of the degree structure, that it would be the other way around, with DCN2, DCN3 at the top and PROG1 at the bottom. This analysis also indicates that programming is a complex activity, even at the level of the first year introductory courses, despite the fact these courses are often regarded by academics as simple and straightforward. This study substantiates Lister's (2003) assertion that in teaching Programming "we have traditionally focused on the higher levels of the taxonomy and ignored the lower levels" (p147).

It is not uncommon when comparing results across a range of courses to encounter significant differences among the same cohort of students. Sometimes these differences are difficult to explain and investigations are instigated to ascertain the cause. A correlation analysis between failure rates and the cognitive difficulty of different courses could be used in such contexts. There may also be potential for using a Bloom analysis in order to standardise results across a range of courses in a similar fashion to diving competitions where the score in a dive is standardised according to its difficulty level.

On a cautionary note it should be observed that at any level of the Bloom hierarchy there are differences between tasks. For although descriptive tasks are placed at a particular level within the Bloom taxonomy (level two) this does not take account of the difficulty of the content to be described. Similarly at the synthesis level it is manifestly easier to program a relatively simple task than a complex task, even though the cognitive level may appear the same.

Degree planners will need to consider whether this data is sufficiently convincing to indicate a reorientation of assessment requirements of courses as suggested by Sanders and Mueller (2000), to bring the sequence level of courses into a closer relation to the Bloom rating. It would appear that degree, stream, and course planners would need to make exceptional efforts to break out of the cognitive domains that dominate each stream. For although from a general educational perspective it is postulated that a stream of courses should transit from a low Bloom rating to a higher one, it does not appear on this evidence to occur in practice. It appears to be difficult to move assessment demands downwards in the Bloom hierarchy for courses in a stream where activity is typically located at the middle or high end of the range. Similarly for courses that are located in streams that have a high-knowledge content, it appears to be difficult to move onwards to higher cognitive tasks, perhaps because there is too much material to learn. The necessity of covering a high quantity of difficult technical material tends to result in insufficient attention being given to higher-level tasks. Accreditation bodies (for example the Australian Computer Society) expect to see an increase in both breadth and depth as the year level of courses rises (Australian Computer Society, nd). However they do not propose any measures (such as use of a Bloom rating) for determining whether these outcomes are achieved.

5 References

Australian Computer Society: (nd): Guidelines For Accreditation of Courses in Universities at the Professional Level. http://www.acs.org.au/ Accessed 3rd Sept 2002.

Bloom, B. S. (Ed.). (1956): *Taxonomy of Educational Objectives Handbook 1: Cognitive Domain*. New York: Longman, Green & Co.

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, Appendix 4. 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. *Proc. Twenty-fifth SIGCSE Technical Symposium on Computer Science Education*, Philadelphia.

Lister, R. (2001): Objectives and Objective Assessment in CS1. *Proc. Thirtieth SIGCSE Technical Symposium on Computer Science Education*, Charlotte, NC, USA.

Lister, R. and Leaney, J. (2003): Introductory Programming, Criterion-Referencing and Bloom. *Proc. Thirty-fourth SIGCSE Technical Symposium on Computer Science Education*, Reno, Nevada, ACM Press.

Reynolds, C. and Fox, C. (1996): Requirements For a Computer Science Curriculum Emphasising Information Technology Subject Area: Curriculum Issues. *Proc. Twenty-fifth SIGCSE Technical Symposium on Computer Science Education*, Philadelphia

Sanders, I. and Mueller, C. (2000): A Fundamentals-based Curriculum for First Year Computer Science. *Proc. Thirty-first SIGCSE technical symposium on ComputerScience Education*.