OVER CONFIDENCE AND CONFUSION IN USING BLOOM FOR PROGRAMMING FUNDAMENTALS ASSESSMENT

TECHNICAL REPORT 681

RICHARD GLUGA, JUDY KAY, RAYMOND LISTER, SABINA KLEITMAN, TIM LEVER

SEPTEMBER 2011
Over Confidence and Confusion in using Bloom for programming fundamentals assessment

Richard Gluga
The University of Sydney
NSW 2006 Australia
richard@gluga.com

Judy Kay
The University of Sydney
NSW 2006 Australia
judy.kay@sydney.edu.au

Raymond Lister
University of Technology Sydney
NSW 2006 Australia
raymond@it.usyd.edu.au

Sabina Kleitman
The University of Sydney
NSW 2006 Australia
sabinak@psych.usyd.edu.au

Tim Lever
The University of Sydney
NSW 2006 Australia
tim.lever@sydney.edu.au

ABSTRACT
A computer science student is required to progress from a novice programmer to a proficient developer through the programming fundamentals sequence of subjects. This paper deals with the capturing and representation of learning progression. The key contribution is a web-based interactive tutorial that enables computer science tutors and lecturers to practice applying the Bloom Taxonomy in classifying programming exam questions. The tutorial captures participant confidence and self-explanations for each Bloom classification exercise. The results of an evaluation with 10 participants were analyzed for consistency and accuracy in the application of Bloom. The confidence and self-explanation measures were used to identify problem areas in the application of Bloom to programming fundamentals. The tutorial and findings are valuable contributions to future ACM revisions, which are expected to have a continued emphasis on Bloom [9].

Categories and Subject Descriptors
K.3 [Computers & Education]: Computer & Information Science Education – Computer Science Education

General Terms

Keywords
programming, Bloom, maturity, competence, confidence, confidence biases, learning progression, assessment, pedagogy.

1. INTRODUCTION
Benjamin Bloom himself once said that the original Bloom Handbook [2] was “one of the most widely cited yet least read books in American education” [12]. The taxonomy is a behavioral classification system of educational objectives. The framework specifies 6 categories, namely, Knowledge, Comprehension, Application, Analysis, Synthesis and Evaluation. Knowledge is the simplest behavior, with each category thereafter being more sophisticated. That is, a Knowledge level objective or assessment task requires a student to simply recall information from memory. In contrast, a Synthesis level task requires students to apply what they have learnt to create new and unique works. A very brief description of the categories follows (adapted from [12]):

- **Knowledge** – recalling of information
- **Comprehension** – interpreting, translating or reordering of concepts, applying a given abstraction
- **Application** – identifying an appropriate abstraction to solve a problem without being prompted
- **Analysis** – breaking down a problem or communication into parts and identifying the relationships between the parts
- **Synthesis** – Identifying and putting together abstractions to create a new and unique artifact or solution to a non-trivial problem
- **Evaluation** – Commenting on the validity of a work with respect to implicit or explicit criteria

The ACM Computer Science Curriculum [10] specifies learning objectives based on the revised Bloom Taxonomy [1]. As an example, the Programming Fundamentals / Data Structures knowledge area specifies the following 9 learning objectives (the italicized verbs are indicative of the Bloom levels):

- **Describe** the representation of numeric and character data.
- **Understand** how precision and round-off can affect numeric calculations.
- **Discuss** the use of primitive data types and built-in data structures.
- **Describe** common applications for each data structure in the topic list.
- **Implement** the user-defined data structures in a high-level language.
- **Compare** alternative implementations of data structures with respect to performance. **Write** programs that use each of the following data structures: arrays, strings, linked lists, stacks, queues, and hash tables.
- **Compare** and contrast the costs and benefits of dynamic and static data structure implementations.
• Choose the appropriate data structure for modeling a given problem.

These objectives show a spread of competence levels ranging from Bloom Knowledge (describe) to Bloom Synthesis and Evaluation (write, implement, compare & contrast). The ACM CS 2013 curriculum is also expected to use Bloom’s Taxonomy, but will likely use a simplified version consisting of only 3 categories: Knowledge, Application and Evaluation [9].

Using Bloom’s Taxonomy to classify objectives and assessment items is not straightforward. For example, Oliver, D. and colleagues [6] invited 4 teaching academics to categorize some assessment questions on the original Bloom Taxonomy scale. For the single example question presented in the paper, the 4 participating lecturers each came up with a different Bloom classification, ranging from Knowledge to Analysis. Whalley, J. and colleagues [8] found the use of Bloom’s taxonomy for rating the cognitive complexity of programming MCQ’s “challenging even to an experienced group of programming educators.” They attributed the difficulty to either some deficiencies in Bloom, or “the authors current level of understanding of how to apply the taxonomy.”

Thompson and colleagues [7] attempted to contextualize the revised Bloom Taxonomy to computer science. They ran an experiment where 5 participants were asked to analyze 6 first-year computer science final exam papers and categorize each question on the Bloom scale. The results showed significant disagreement between the rankings performed by different participants. This was attributed to some having implicit knowledge of how the subject was taught, and hence having a better understanding of the cognitive processes of the students undertaking the exam papers. Thompson and colleagues [7] however did not discuss the participants’ prior knowledge of the Bloom Taxonomy, or its application in a computer science context. It was only after the participants had a chance to collaborate and discuss each classification that they reached consensus on each exam question.

On the basis of our reading of the above work, we concluded that academics seeking to use Bloom’s Taxonomy need to be trained. To this end, our contribution described in this paper is a computer science contextualized web-based tutorial on the Bloom Taxonomy with interactive examples, user self-explanation and self-reflection. The tutorial is a useful resource in training participants on the application of Bloom in classifying programming assessment questions. The results from the evaluation of this tutorial are useful in identifying where Bloom is used inconsistently due to different assumptions about the learner, different interpretations of the Bloom categories, or a misunderstanding of the categories. The tutorial and Bloom insights are important inputs to future work on measuring learner progression in computer science and future ACM CS revisions.

2. METHOD

ProGoSs (Program Goal Progression) is an online web-based system built as part of our research on curriculum mapping and measuring learning progression in higher education. In this experimental setup, participants were invited to login to the system and take our interactive Bloom tutorial. A detailed discussion on the design and content of this tutorial is provided in [21]. The following is an overview of the tutorial structure that is sufficient for the evaluation results discussed in this paper.

The tutorial commenced with a pre-survey asking the participants to self-rate their own confidence, based on their existing knowledge, in being able to correctly classify programming exam questions on the Bloom scale. We will refer to this confidence judgment as the Initial Confidence score (IC), which was measured as a percentage, with 100% indicating complete confidence.

After the pre-survey, participants were asked to read a description of each Bloom Taxonomy category, contextualized with wording relevant in programming. The categories were introduced as shown in Figure 1.

![Figure 1: Tutorial Bloom Categories](image)

Each category was described and illustrated with an example in a tab-based layout as seen in Figure 2.

![Figure 2: Tutorial Category Description](image)

After reading each description and example, the participants had to self-rate their confidence in being able to apply the Bloom category in classifying programming questions. We call these the Prediction Confidence (PC) scores. These were measured as a percentage for each Bloom category. Participants moved from one tab to the next to read and rate each category.

After reading the 6 Bloom category descriptions, and self-rating their understanding of each, participants were then asked to classify some examples of examination questions. Participants had to provide answers, explanations and ratings on each of the 12 examples, such as the example seen in Figure 3. Participants are encouraged to scroll back to refer to the category definitions if needed. A progress bar fills up as they complete each example. For each example, the participants classified the exam question on the Bloom scale. Participants were then required to self-rate their confidence in their classification (On-Task Confidence), as well as to justify answers and comment on any uncertainties in their confidence. This was done in accordance with work by Chi M.T.H and colleagues [13] showing that “Eliciting self-explanations improves understanding”.

The 12 example questions, and earlier category descriptions, were created by our Bloom expert - a computer science academic with
an active research interest and publication record in the application of Bloom to programming. Out of the 12 example questions, 3 were targeted as Knowledge, 2 as Comprehension, 2 as Application, 2 as Analysis, 1 as Synthesis and 2 as Evaluation. The uneven numbers were used so that participants would not be able to guess the last few by discerning the pattern and counting answers. The order in which the questions were presented was random, with the same sequence used for all participants.

The experiment took 60 minutes to complete. Results indicate that for each category On-Task Confidence judgment was higher than Prediction Confidence judgment. This indicates that participants felt more capable about applying the Bloom taxonomy after undergoing the Bloom tutorial.

### Results

3. **RESULTS**

#### 3.1 Participants

A total of 10 participants completed our interactive tutorial and assessment classifications. These consisted mostly of computer science tutors and 1 computer science professor. The experiment took 60 minutes to complete.

#### 3.2 Initial Confidence

The average participant Initial Confidence (IC) score was 30%. This is the confidence of participants in being able to apply Bloom in classifying programming questions before taking our tutorial.

#### 3.3 Prediction Confidence, On-Task Confidence and On-Task Accuracy

The average participant Prediction Confidence (PC) scores after reading our initial tabbed Bloom category descriptions ranged between 67% and 71%, as seen below. The results after participants had classified all 12 example questions are summarized in Figure 4. The chart shows the 6 Bloom categories along the horizontal axis. Each category is subdivided into 3 columns. These are, from left to right, Prediction Confidence, On-Task Confidence and On-Task Accuracy. The On-Task Accuracy is the percentage of participants who agreed with our expert classification for each question, i.e. answered correctly.

Results indicate that for each category On-Task Confidence judgments had a more homogeneous range (between 77% for the Analysis and 90% for the Comprehension categories) than the relevant On-Task Accuracy scores (ranging between 53% for the Knowledge and 90% for the Evaluation categories). However, for only two categories – Evaluation and Analysis – the On-Task Confidence levels matched the On-Task Accuracy scores (their difference, referred to as Bias scores, were -3% and 2% respectively). In all other categories, a different degree of overconfidence (confidence exceeds the actual accuracy) is evident: 28% (for Knowledge), 20% (for Synthesis), 13% (for Application) and 10% (for Comprehension). This indicates that, despite their subjective feeling of confidence, participants had difficulty classifying correctly the items for these four categories, especially Knowledge and Synthesis. Importantly, the high degrees of overconfidence for the Knowledge category strongly suggests that these examples, and/or the use of this Bloom category, may also be tricky or misleading (see [14][15][16][17] for reviews), suggesting a need to reconsider the selection of examples that evaluate this category in future studies. The use of On-Task Confidence judgments provided the novel and additional information allowing such diagnostics.

#### 3.4 Confusion Matrix

Figure 5 is a confusion matrix that shows all of the participant classifications for each example. Along the left as rows is the expert classification. Along the top are the actual participant classifications. Each cell has the question number, followed by a colon and then by a character code that indicates if the participant:

- was correct (C);
- incorrect but agreed with our expert explanation after being shown the answer (A);
- incorrect but still disagreed with our explanation after being shown the answer (D);
- incorrect and did not agree or disagree with our explanation after seeing the answer (N).

For example, question 11 is an Application question. Five participants agreed with our expert, as shown in the
Application/Application cell (11:CCCCC). Of the other 5 participants, 1 thought it was a Knowledge question and disagreed with our explanation (11:D), 1 thought it was Comprehension and neither agreed or disagreed with our explanation (11:N), 1 thought it was Analysis and disagreed (11:D), and 2 thought it was Synthesis but then agreed with our explanation (11:AA).

This was caused by example question 9 in particular (discussed below).

The tutorial confidence matrix shows every participant classification for every example and how it differed from our expert classification. Again the biggest discrepancy here appears in the Knowledge row, namely for example 9. Our expected classification for this was Knowledge, yet no participants picked this. The participant classifications were 4 Synthesis, 5 Application and 1 Analysis. Example 9 was a question that asked students to write a SortedMap implementation “as discussed in lectures and practiced in tutorials.” Since the question suggests that students have had repeated practice at this particular exercise, it can be argued (as our expert did) that it is a Knowledge level task. From reading the participant feedback comments, 5 agreed with this explanation after being shown the answer, 3 did not comment, and 2 disagreed on the basis that a student would not be able to memorize that much code and would thus have to pick abstractions to implement the algorithm (and hence why these 2 participants picked the Application level). The example highlights the tight dependence on knowledge of the teaching context to correctly classify exam questions using Bloom. That is, different assumptions about the teaching context can lead to vastly different classifications.

Additionally, the disagreements (D) in the confusion matrix stemmed mostly from one participant. He had a somewhat higher than average Initial Confidence (40%), perhaps bringing preconceptions about Bloom and its use in this context. The self-explanations left by this participant gave well-reasoned arguments. For example, our expert coded the task to write a function that returns the minimum value in an array an Application; the student solution should traverse an array and update the minimum, both very familiar and practiced elements by the end of a second programming subject. Importantly, the is little potential for diverse, correct solutions. However, this participant appeared to take the perspective of a very novice programmer, and argued it was a Synthesis task, as the student needs to think about many abstractions and put them together into something that is not immediately obvious to them.

### Figure 5: Confusion Matrix
The shaded cells represent correct answers, that is, participants made the same classification as our expert. The thick-bordered cells represent the break-down of Bloom into only 3 distinct categories. That is, combining Knowledge with Comprehension, Application with Analysis and Synthesis with Evaluation. This banding is significant as it is the proposed framework for the ACM CS 2013 revised curriculum [10]. This highlights participants who answered incorrectly, but were still within the correct band grouping. Participant answers outside of the thick-bordered cells are undesirable, as they represent a greater distance in classifications (i.e. more severe inconsistency).

### 3.5 Final Confidence and Participant Feedback
After completing the 12 examples, the average participant Final Confidence score was 75% (an increase from the 30.8% Initial Confidence before starting the tutorial). This non-trivial increase of 44.2% of confidence could be attributed largely to the training they experienced during the Bloom tutorial.

All participants responded positively in the final feedback question. The common trend in these comments was that the category descriptions were good for gaining a basic grasp, but the interactive examples with justified answers were very useful in solidifying their understanding.

### 4. DISCUSSION
Most of our participants had very little exposure to Bloom prior to taking the tutorial, hence the low average Initial Confidence of 30.8%. Only 3 participants had an Initial Confidence greater than 30%. These were a computer science lecturer, an educational researcher and a computer science postdoc. After reading our descriptions of the Bloom categories, participant Predictive Confidence increased to around 70% across all Bloom categories. This was a good indicator that the category descriptions were helpful to those with very limited prior knowledge.

The chart in Figure 4 shows participants were on average overly confident in their answers however. The largest difference between confidence and accuracy is in the Knowledge category.
that a seemingly Synthesis level task for one programmer, may be
more like an Application or Knowledge level task for another.
That is, writing a function to, say, return the minimum value in an
array, may require a lot of thought and problem solving ability for
a novice who derives a solution from first principles. For an
experienced programmer however, the solution may be strikingly
obvious and require very little thought. A method to measure this
hypothesis could be a useful tool in assessing the competence
level of a programmer.

Our work was motivated by the wide use of Bloom in defining
curriculum, particularly in the case of programming fundamentals
[10] and our concerns about the reliability with which it was being
used. This paper makes a contribution in presenting an online
Bloom tutorial and reporting on its evaluation. We believe that
our ProGoSs system's Bloom tutorial is the first such system that
helps teachers of programming fundamentals have greater
understanding of Bloom, as a foundation for more systematic
design of teaching and learning materials and assessment of how
well student learning meets the intended goals.

6. ACKNOWLEDGMENTS
Will insert in camera-ready version if accepted.

7. REFERENCES
W., Cruikshank, K. A., Mayer, R. E., Pintrich, P. R., Raths,
J. and Wittrock, M. C. (2001). A taxonomy for learning,
teaching and assessing: A revision of Bloom's Taxonomy of

Objectives: Handbook I: Cognitive Domain, Longmans,
Green and Company.

learning of generic skills. In V. Alevon, J. Kay, and J.
Mostow, editors, ITS2010, Proceedings of the Tenth
International Conference on Intelligent Tutoring Systems,

models supporting flexible curriculum definition. In
Proceedings of the Workshop on Scalability Issues in AIED,

and its Blooming Assessment. Proc. Fourth Australasian
Computing Education Conference (ACE2000), Melbourne.
pp 158-162.

This course has a Bloom Rating of 3.9. Proc. Sixth
Australasian Computing Education Conference. Volume 30
(ACE '04), Raymond Lister and Alison Young (Eds.), Vol.
30. Australian Computer Society, Inc., Darlinghurst,
Australia, Australia, 227-231.

Proc. ACE2008 Tenth Conference on Australasian
Computing Education. Vol. 78

Study of Reading and Comprehension Skills in Novice
Programmers, using the Bloom and SOLO Taxonomies.
Eighth Australasian Computing Education Conference
(ACE2006), Hobart, Australia. CRPIT, 52, pp. 243-252.

http://www.118.ifip.info/wise/wise7/resources/cs/2013.ppt

[10] ACM Computer Science Curriculum 2008,
http://www.acm.org/education/curricula/ComputerScience20
08.pdf

Going SOLO to assess novice programmers. Proceedings of
the 13th annual conference on Innovation and technology in
40.

taxonomy: a forty-year retrospective. Chicago: University of
Chicago Press


Cognitive Sciences, 1 (2), 78-82.

consequence of informal experimenter-guided selection of
almanac items. Organizational Behaviour & Human
Decision Processes, 57 (2), 226-246.

Probabilistic Mental Models - a Brunswikian theory of

oriented aspects of metacognitive processes in test-taking.

[18] Centre for the Advancement of Teaching and Learning,
University of Western Australia (2005). A Basic Guide to
Writing of Student Outcome Statement. Retrieved 24 August
2011 from

[19] Centre for Learning and Professional Development,
Retrieved 24 August 2011 from

Writing learning outcomes. Retrieved 24 August 2011 from
http://www.mq.edu.au/lrc/about_lrc/assess_docs/writing_learn
_out.pdf

Coming to terms with Bloom: an online tutorial for teachers
of programming fundamentals. Proc. Fourteenth Australasian
Computing Education Conference (ACE2012), Melbourne.
Pp. TBA