1. INTRODUCTION
When writing exams, many computing educators find themselves wishing there existed repositories from which they could just copy exam questions. Such repository sites exist, but most of those sites do not have a lot of exam questions in them. While copying exam questions from a repository has its obvious attractions, there is little motivation for most academics to put exam questions into such repositories.

This project proposes to reverse the repository building around. Instead of building the infrastructure for a repository in the hoping that “if we build it they will come”, this project will focus on producing content that repository builders would then be free to upload. The questions written in this project will also be made available in static documents, so academics can simply read through those documents to find useful questions.

2. Why MCQs?
Many CS1/CS2 educators might prefer a collection of code writing questions, but we believe that developing a set of multiple choice questions is the best place to start, for several reasons, including:

- MCQs are very commonly used, especially when class sizes are large, because they are easy to grade.
- The use of MCQs in connection with clickers and online test delivery systems is becoming common.
- Educators are more likely to use a battery of MCQs than a battery of code writing questions, as many educators do not enjoy writing MCQs.
- There is an extensive literature on what comprises a good MCQ (See overview in Lister et al. 2004).

3. Why 1024?
Approximately one thousand MCQs should provide enough variety of MCQs, and topic coverage, to elicit interest from educators and repository builders. (And writing 1,048,576 MCQs seemed a little too ambitious.)

Achieving our 1K target is not essential. That achievement depends in part on the size of the working group. The quality of questions will not be compromised in order to reach the 1K target.

4. Coverage
We aim to provide MCQS that are suitable for students who are at any point in their first twelve months of learning to program. If that definition seems too vague, readers might examine the “Software Development Fundamentals” knowledge area in CC2013 Ironman (ACM/IEEE-CS Joint Task Force, 2012). However, the working group leaders do not regard CC2013 as prescriptive: we are aware of the wide variety of topics commonly taught in the first year of programming – if a particular MCQ is suitable for students in the first year at your institution, then it is probably a suitable MCQ for this project.

Both formative and summative question are welcome. Questions need not be aimed at final exams, or even mid-semester tests. Questions suitable for weekly quizzes are welcome, as are questions that might be used as “clicker questions” during lectures.

Questions may be written in any programming language that is reasonably commonly used to teach novices in their first year. However, questions should in principal be translatable into other languages.

Questions need not be language dependent. Questions using pseudocode and questions testing conceptual knowledge are also welcome.

Questions for both the imperative and object-oriented paradigms are particularly welcome, and equally welcome. No paradigm is unwelcome, provided the particular language used in an MCQ is reasonably commonly used to teach that paradigm.

5. Metadata: Classification of Questions
Contributed questions will need to be accompanied by metadata, so that the 1K questions are well indexed by web search engines. We envisage that anyone looking for an MCQ on, say, a particular topic should be able to do so by placing appropriate search terms into a search engine.

The starting point for the classification scheme is Simon et al’s (2012) scheme, which includes for example:

- **Topics covered:** for example data types & variables, constants, strings, I/O, file I/O, GUI design and implementation, error handling, program design, programming standards, testing, scope (includes visibility), lifetime, OO concepts (includes constructors, classes, objects, polymorphism, object identity, information hiding, encapsulation), assignment, arithmetic operators, relational operators, logical operators, selection, loops, recursion, arrays, collections (other than arrays), methods (includes functions, parameters, procedures and subroutines), parameter passing, operator overloading.
• **Skill required**: includes knowledge recall, tracing code, identifying the purpose of code, and “writing code” (by selecting appropriate lines to complete code). See Simon et al’s (2012) for a longer list. If additional skills are identified during the project, those new skills will be added.

• **Intellectual complexity**: where the question fits in Bloom’s taxonomy. All working group participants will be required to complete an online tutorial about Bloom’s taxonomy, so that the participants use Bloom’s Taxonomy consistently. The tutorial takes 1-2 hours.

Simon et al’s (2012) classification contains other classification areas that we do not cover here. See also the overlapping classification scheme reported in B. Simon et al. (2009).

6. **Phase 1: Collaborative Authoring**

The production of the MCQs will involve at least two phases. In the first phase, MCQs will be written, tagged and vetted for suitability. This first phase is discussed in this section. The second phase will be the final delivery of the MCQs and metadata. That second phase is discussed in the next section.

The first phase will involve quite a lot of data sharing and communication among the working group participants, most of which will need to happen prior to meeting at the conference. The working group leaders have identified Peerwise (Denny et al., 2008) as a possible vehicle for hosting this first phase:

http://peerwise.cs.auckland.ac.nz/

7. **Phase 2: Final Delivery of MCQs/Metadata**

As the aim of this project is NOT to build repository software, but instead to supply content for repositories; the working group will therefore use a low-tech approach to final delivery of selected questions.

Participants will provide their selected MCQs and metadata in human readable form, in plain text or via word processing products such as (but not exclusively) Microsoft Word. Any commonly used document format and easily/freely obtained format reader should be okay, but the working group leaders will make the final decision on suitability, on a case-by-case basis.

The exact format of MCQs and their metadata in these documents will not be specified precisely. However, the format will be specified to sufficient degree so that:

1) Anyone looking for an MCQ on, say, a particular topic should be able to do so by placing appropriate search terms into a web search engine.

2) A repository developer should be able to write software to parse the MCQs and associated metadata, for upload to their repository.

8. **Working Group Membership Criteria**

To qualify for participation in the working group at the conference, people must either:

1) Contribute a minimum number of acceptable multiple choice questions prior to meeting at the conference. The exact value of that minimum number is a tradeoff between quantity and quality, and the setting of that number will be an early item for discussion among potential working group members.

(2) Perform some other function, such as classifying or vetting submitted MCQs.

While the criteria are currently flexible, and the final criteria will be negotiated, the working group leaders will be the final judges on who has met the criteria.

9. **SIGCSE 2013**

People interested in joining the working group, but who first need to discuss it further, are welcome to email Raymond and Kate. People attending SIGCSE 2013 might prefer to have a face-to-face conversation with us. If so, you still may want to email us prior to SIGCSE 2013, and set up a meeting time and place.

10. **REFERENCES**


