Abstract—This Work-In-Progress Research Paper reports on an international study that is being undertaken in order to develop a validated concept inventory for the second introductory computer science course (CS2).

A concept inventory is a research-based multiple-choice test that measures a student’s knowledge of a set of concepts while also capturing conceptions and misconceptions they may have about the topic under consideration. Development of a concept inventory for a course requires identifying course topics that are both difficult and important. This paper details how the Delphi method is being used to develop a concept inventory for CS2; in particular, it focuses on the initial process of identifying the set of topics that should be covered by a concept inventory for CS2.

Index Terms—Concept inventory, Delphi method, Computer Science 2, CS2, Assessment

I. INTRODUCTION

Almost every computer science program contains two semester-long introductory courses, usually named Computer Science 1 (CS1) and Computer Science 2 (CS2). While CS1 focuses on basic coding and computational thinking, CS2 moves beyond basic programming skills to an exploration of data structures, algorithms, and design. A quick literature review quickly reveals that there is a huge numerical preponderance of articles that focus on CS1 [1] [2] [3] [4] rather than on CS2; though aspects of CS2 have begun to receive more attention in the literature in recent years [5], [6], [7]. This may be because there is broad agreement on the key elements of CS1, while the curriculum for CS2 can vary significantly between institutions [8].

It can be argued that the intent behind the teaching of CS2 is open for debate – is its purpose to acquaint students with a standardized set of tools and frameworks, or is it to give them an appreciation of the design of such frameworks and the implementations underpinning them? One phenomenographic [9] study of this question [10] identified five categories of intent in the teaching of CS2: “developing transferrable thinking, improving students’ programming skills, knowing ‘what’s under the hood’, knowledge of software libraries, and component thinking”. Regardless of the purpose behind the teaching of CS2, it has been a mandatory element of the ACM Computing Curriculum since 1978 [11] and is likely to remain so for many years to come.

Most undergraduate courses, including CS2, come with a clearly identified set of prerequisites that ensure that those taking the course have the skills needed to successfully navigate and complete it. Underpinning these prerequisites is an expectation that students have acquired a set of preconceptions that enable them to develop a deeper understanding and appreciation of the concepts they will meet in the course of their studies. However, many students also enter each course with misconceptions that can impede their progress [12] and led them to become frustrated and disengaged. In order to address these deficiencies, instructors must first identify them and then develop classroom interventions and strategies that address them [13]. One way of doing this is through the use of a concept inventory.

A concept inventory (CI) seeks to determine how well each student’s conceptual framework matches an accepted conceptual framework of the given subject [14]. This is achieved using a research-based multiple-choice test where each question includes one correct answer and a set of incorrect answers that result from misconceptions of the topic.

This work in progress reports on an international study [15], [16] that is being undertaken in order to develop a concept inventory for CS2. When the authors’ embarked upon this study in 2017, there had been little progress towards the development of a concept inventory for CS2, since then once such concept inventory has been developed [7]. It focused solely on North American institutions and the six authors drew on a small expert panel of nine faculty members drawn from a range of institutions. In addition, the study drew on interviews with 50 students from 3 separate institutions. By contrast, the study reported on in this paper draws upon an international pool of expert faculty members and is closely following a Delphi process [17], a well-established process for creating concept inventories [14], [18], [13], [19]. Once the study reported on in this paper is complete, it will broaden
the understandings developed in [7].

II. PROGRESS

A. Methodology: the Delphi method

In order to identify the topics for inclusion in the CS2 concept inventory, this study uses the Delphi method. The Delphi method is a systematic iterative process of arriving at a common opinion or decision by a diverse group of experts [20]. By seeking consensus amongst the expert group, the Delphi method is designed to eliminate the natural bias that any individual expert will have based on their experience and field of expertise [13]. At each round, the experts are provided with a questionnaire and an aggregate group response at the preceding round. The latter may affect the experts’ responses at a current round. After several rounds, the group’s response closely reflects the consensus opinion of the group. By giving each expert’s opinion equal weight and through the use of anonymous feedback, each expert is only influenced by the opinions and arguments put forward by other experts, and not by their reputation in the field [21].

In this study, the experts are drawn from a pool of experienced CS2 instructors. They were first asked to identify C2 topics that have relatively high importance in the course and are difficult for the students to master. After that they rank the topics by difficulty and separately, by importance, using the feedback of other experts aggregated by the researchers. Two more ranking refinement rounds follow as outlined in Figure 1 and detailed in the rest of this section.

B. Preliminary step: Recruiting and selecting experts

We recruited the experts through a combination of emails, advertising at conferences, and networking. Emails went out to over 400 academics teaching CS2 worldwide, identified by scanning various university websites. We also sent emails to CS2 textbook authors and academics who published CS2 related research. We gave a Lightning talk [16] and also reached out to our network of contacts. Diversity was a key point in our recruiting attempts. We actively recruited to get academics from different continents (North America, South America, Asia, Oceania (New Zealand, Australia), Europe, and Africa). Overall, we enlisted 34 initial experts.

The initial experts filled out a survey and described the major portion (60% or more) of their CS2 as an introduction to either object-oriented programming or data structures. The data structures choice was further divided into two categories: using the data structures provided by libraries/APIs and implementing the data structures (and likely using them too). 25 of the initial experts returned the completed survey. Of those, 19 experts indicated the category of data structures and implementation (our concept inventory target).

We sent out an envelope information survey to those 19 “data structures and implementation” experts and received 18 responses. 17 of the responders identified as educators in the classroom, 6 as researchers in CS educational topics, one was a textbook author, and one had taught CS2 recently but not currently; several experts selected more than one role. All of our experts were at 4-year institutions; 8 at institutions with large graduate programs and 10 at primarily undergraduate institutions. 17 were at institutions in North America and 1 in Asia. 3 were female and 13 were male. We have achieved some diversity with respect to gender, institution level (graduate and undergraduate), geographic location, and role (teachers, researchers, textbook authors).

C. Round 1: Topic set generation

For Round 1 of our application of the Delphi method, each expert generated a list of 10-15 topics they considered to be both difficult and important in CS2 (the leftmost box in the top row of Figure 1). They were asked to consider their entire CS2 course, not just the portion focused on data structures. 17 experts completed this step.

When analyzing Round 1 responses we found that the joint topic set was very large. We also observed that the judgment whether two answers were describing essentially the same

![Fig. 1. Delphi method workflow for identifying the topics for CS2 concept inventory. The steps outlined in top and bottom rectangles in each column constitute one round of the Delphi method.](image-url)
topic or two different topics was very subjective. Our initial plan for the analysis portion of Round 1 was to produce a joint set of topics mentioned by 2 or more experts. However, we wanted to preserve the diversity of opinions that the experts had to offer. Additionally, there could have been topics considered by several experts that only one expert chose to include in the list at this round. For these reasons, we decided to keep all topics mentioned by the experts in the joint set of topics (the leftmost box in the bottom row of Figure 1). Topics that are of true interest only to one expert will be eliminated after Round 2 is completed.

After removing duplicate answers, the joint set has 120 topics. To aid the experts in ranking them at Round 2, they were partitioned into 16 categories as shown in Table 1.

### D. Round 2: Ranking the topics by difficulty and importance

At Round 2 experts are asked to rank the topics from the joint set by difficulty and, separately, by importance (the second box in the top row of Figure 1). Each expert received a spreadsheet file of the 120 topics organized into 16 categories. The order of categories and the topics inside each category were randomized for each expert individually to prevent the ordering biases from influencing the results. The ranks such as "80th most important topic" or "120th most difficult topic" are both hard to determine and of little use for the purpose of the study. Therefore, the experts were asked to select 30 of the most difficult topics and rank them by difficulty, and separately, select 30 of the most important topics and rank them by importance, and then to return these two ranked lists.

As of April 2020, 9 experts have returned their Round 2 lists. Two experts have indicated that due to issues handling the COVID-19 pandemic, they will not be able to complete phase 2 but hope to rejoin the study at a later date. The remaining 6 experts are still working on Round 2.

### III. CURRENT OBSERVATIONS

Analysis of Round 2 responses makes it clear that there isn’t a single topic that all experts agree is important. Figure 2 shows ten of the most frequently occurring topics in the selections of the 9 experts who completed Round 2. Binary search (Searching category) and Hash tables (Map and Hash Table category) have the most occurrences, with 7 of 9 experts putting them in their list of top 30 most important topics.

It is of note that there is no single topic that all experts agree is challenging. Figure 3 shows ten of the most frequently occurring difficult topics in the difficulty rankings. Analysis of Recursive Algorithms (Analysis and Big-O category) has the most occurrences with 7 of 9 experts putting it in their top 30 most difficult topics. Hash tables and Recursion (Recursion category) appear in both top 10 lists making them the two topics that experts consider both difficult and important.

Figure 4 shows the relative importance of the topic categories. Each category was given a “popularity” score reflecting how often a topic was selected from this category.

\[
\text{popularity} = \frac{\text{selection_count}}{n \times \text{category_size}}
\]

*selection_count* is the number of times the topic was selected by different experts, *category_size* is the number of topics in this category, and *n* is the number of participating experts. Experts found Heaps, Recursion, and Searching to be the most important categories. Figure 5 illustrates the relative difficulty of the topic categories. The experts found Recursion, Heaps, and Analysis & Big-O to be the most difficult categories. It is interesting to note that Heaps and Recursion are on or near the top in both metrics, suggesting that Heap and Recursion

### TABLE I

<table>
<thead>
<tr>
<th>Topic Categories</th>
<th>Array</th>
<th>Graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstraction and ADT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis and Big O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Map and Hash Table</td>
<td>Heap</td>
<td></td>
</tr>
<tr>
<td>Recursion</td>
<td>List</td>
<td></td>
</tr>
<tr>
<td>Searching</td>
<td>Memory</td>
<td></td>
</tr>
<tr>
<td>Software Engineering</td>
<td>Set</td>
<td></td>
</tr>
<tr>
<td>Sorting</td>
<td>Tree</td>
<td></td>
</tr>
<tr>
<td>Stack and Queue</td>
<td>Other</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 2. Top ten most important topics as identified at Round 2 of Delphi method application ordered by the number of occurrences in experts’ importance rankings

Fig. 3. Top ten most difficult topics as identified at Round 2 of Delphi method application ordered by the number of occurrences in experts’ difficulty rankings

Fig. 4. Relative importance of the topic categories. Each category was given a “popularity” score reflecting how often a topic was selected from this category.
Based on the preliminary findings of the first two rounds of the Delphi process, we note that such CS topics as recursion, heaps, parallelism, sorting and searching, big-O notation and asymptotic algorithm analysis, linked lists, queues, maps and hashing are among most difficult and important ones in the CS2 courses (Table II). These are the topics that merit further research and that we anticipate our resulting concept inventory will focus on.

### REFERENCES


