Community Input and Engagement for CS2023: Foundations of Programming Languages

Michael Oudshoorn
High Point University
High Point, North Carolina, USA
moudshoo@highpoint.edu

ABSTRACT

Currently, an ACM/IEEE-CS/AAAI task force is undertaking a review of the computer science curriculum guidelines [7]. The task force is aiming to generate an initial draft, known as CS2023, by the end of 2023. It is anticipated that the resulting curriculum guidelines will impact computer science curriculum globally until at least the year 2034. This curriculum review process is undertaken every decade or so, and was last completed in 2013 with the resulting ACM/IEEE-CS 2013 document [1]. CS2013 has served the community well and planned revisions focus on the removal of topics that are no longer relevant, identification of topics that are new or have increased importance as we look toward 2034 and the end of the lifespan of this new document. An integral part of the review process is to solicit input from a wide audience. While reviews can be made on-line, the purpose of this Birds-of-a-Feather (BOF) session is to present the proposed changes to the programming languages knowledge area and to solicit input from the attendees. There are some substantial changes being proposed and timely feedback from the community is essential. Two of the major changes are the migration to a competency model, similar to that described in CS2020 [2], and the replacement of Tier-1 and Tier-2 hours with a CS core that all computer science student should know, and a a KA core which students specializing in the knowledge area should know.

CCS CONCEPTS

• Applied computing → Education; • General and reference → Computing standards, RFCs and guidelines; Reference works; • Theory of computation → Formal languages and automata theory; Semantics and reasoning; • Software and its engineering → General programming languages: Formal language definitions.

KEYWORDS

curriculum, programming paradigms, programming languages, formal methods, language translation

ACM Reference Format:

1 INTRODUCTION

Programming languages, models of computation, and formal methods are a key component of every computer science program, and has been a knowledge area in every revision of the computer science curricula guidelines since its first appearance in 1969 [1, 3–6, 8]. An ACM/IEEE-CS/AAAI task force has been established with revising the current computer science curriculum guidelines [1] with a goal of publishing the revised curricula guidelines in late 2023 [7]. It is anticipated that the new guidelines will be in place for at least a decade before yet another revision takes place. The programming languages knowledge area is currently being reviewed as part of the CS2023 [7] process and community feedback is an essential and integral part of the process.

2 THE PROPOSED CHANGES: CS2013 TO CS2023

Computer Science has been a rapidly evolving discipline since its inception. This continues to be true today. The changes being proposed to the curricula guidelines include a shift away for the use of Tier-1 and Tier-2 core hours in the current guidelines [1], a movement toward a competency based model of assessment [2], and some changes to the topics/content that comprise each knowledge area (including merging some knowledge areas). In addition, society, ethics and professionalism, including diversity, equity and inclusion, is taking on a larger role within the curricula guidelines, especially as it is being considered a cross-cutting knowledge area.

Specifically for the programming languages knowledge area, the most significant changes include the way in which core hours are defined, the use of a competency model, and the topics that make up the knowledge area. In addition, the knowledge area has been renamed Foundations of Programming Languages to reflect the fact it is focused on programming language paradigms and concepts and not on a specific programming language, or on how to program. The significant changes are described briefly below.

2.1 Core Hours

The CS2023 curriculum guidelines abandons the use of core tier-1 and tier-2 hours for topics. In CS2013 student should have seen 100% of the tier-1 hours, and 80% of the tier-2 hours. In the place of tier-1 and tier-2 hours the notion of CS-core hours, KA-core hours, and elective knowledge units is introduced. One CS-core hour indicates that one hour of class time should be spent on a knowledge unit and that all computer science students should see this topic and gain this knowledge. One KA-core hour indicates
that for those programs that plan to cover the topic in greater depth, that one hour of class time should be spent on a knowledge unit. Hence, a knowledge unit assigned CS-core hours is a knowledge unit that all computer science students should be exposed to, and a knowledge unit assigned KA-core hours represents a topic that only students focusing on the knowledge area will be exposed to. Knowledge units that are designated elective are interesting topics that programs might cover if there is sufficient time.

This change in essence means that the number of CS-core hours is the sum of tier-1 hours plus 80% of the tier-2 hours. Of course, with the evolution of the importance and relevance of some knowledge units have been added to reflect their continuing and growing importance, some contend will be shared with other knowledge units such as Operating Systems, and Architectures and Organization.

Some topics from some knowledge units such as the Parallel and Distributed Computing Knowledge Unit are likely to be integrated within the curriculum with topics from other knowledge areas.

3 SUMMARY

The 2013 version and the current proposed version of the programming languages knowledge area is found at https://csed.acm.org/. Participants at this Birds-of-a-Feather session are encouraged to read the knowledge area description and come prepared to discuss the changes and offer their thoughts on the direction this work is taking.

ACKNOWLEDGMENTS

Thank you to the foundations of programming language knowledge area subject matter experts: Doug Lea, State University of New York at Oswego, NY; Annette Bienen, TU Kaiserlautern, Germany; Alan Dearle, University of St. Andrews, Scotland; Brijesh Dongol, University of Surrey, UK; Michelle Kuttel, University of Cape Town, South Africa; James Noble, Victoria University of Wellington, New Zealand; Peter-Michael Osera, Grinnell College, IA; Michelle Mills Strout, University of Arizona, AZ; Mark Marron, Microsoft Research, WA.

REFERENCES


