

# Workshop: Computational Thinking for Practicing Engineers

Orieta Celiku  
School of Computer Science  
Carnegie Mellon University  
Pittsburgh, PA 15213  
+1 412 268 3501  
orietac@cs.cmu.edu

David Garlan  
School of Computer Science  
Carnegie Mellon University  
Pittsburgh, PA 15213  
+1 412 268 5056  
garlan@cs.cmu.edu

Anthony Lattanze  
Institute for Software Research  
Carnegie Mellon University  
Pittsburgh, PA 15213  
+1 412 268 4736  
lattanze@cs.cmu.edu

## Abstract

*This workshop will address the problem of providing education in foundational computer science principles and techniques for the large population of practicing software developers who do not have a formal computer science background. Participants will be introduced to an NSF-sponsored project within Carnegie Mellon University's Masters in Software Engineering Programs to bridge gaps in formal training of practicing engineers using modules that cover the foundational material in a way that can be put to use directly. A key feature is tight coupling with practical exercises within the context of hands-on applications. Participants are encouraged to contribute ideas regarding content, delivery methods, outcomes, and examples.*

## OBJECTIVES

Advances in the software industry have led to an increasing need for *computer professionals* with solid foundational knowledge in computational thinking [2]. Unfortunately many professionals do not have this training, having entered the field of software development through other disciplines.

Currently suitable educational alternatives do not exist for this population. Taking a large number of undergraduate courses in foundational computer science topics is inappropriate because of the time commitment and the fact that such courses are oriented towards students with little experience or domain expertise. Moreover, most graduate courses typically assume this knowledge as a prerequisite.

This workshop will explore the problem of providing educational alternatives that would allow practicing engineers to acquire computational thinking fundamentals. Through presentations and guided discussion, the workshop will consider the problem and possible solutions, informed by our initial experience in developing an NSF-sponsored bridge curriculum as part of the Masters in Software Engineering Programs at Carnegie Mellon University [1].

## OUTLINE OF THE WORKSHOP

The workshop will be organized in four main parts:

1. Introduction to the problem and requirements for a solution.

In this part of the workshop we will discuss the need for computational thinking skills for modern computer professionals, particularly in addressing modern engineering concerns, such as scalability, performance, and reliability. We will also consider problems with existing educational alternatives in meeting this need. We will lead a discussion on the requirements that a course that can address the needs.

2. Ongoing efforts to bridge the gaps in the formal training of practitioners

In this part of the workshop we will present CMU's initial efforts to develop a bridge course that addresses the gap in educational alternatives for practitioners. Key features of the approach include (a) distilling and packaging the set of core, foundational concepts of computational thinking that have stood the test of time and are relevant broadly to professionals working in software development or other disciplines that have a significant computational component; (b) grounding the material in practical applications so that students can see how these concepts apply in practice; and (c) streamlining of the course content to take advantage of the professional maturity of the students.

### 3. Discussion

In this part of the workshop we will lead a discussion addressing questions such as

- What areas of computer science are most essential and/or poorly understood by practitioners?
- What competency level should the course aspire to?
- How should the presentation of course materials be streamlined to take advantage of the students' experience?
- What kinds of real world examples are good vehicles for conveying the foundational concepts?

### 4. Wrap-up, conclusions, and next steps

## EXPECTATIONS

During the workshop participants will be introduced to this area and exposed to the materials developed at Carnegie Mellon, including course syllabus, readings, and exercises grounded in practical systems. Feedback on this curriculum, and discussion about related experience in the area will be solicited.

We hope that the workshop will generate interest in further discussions and, possibly, lead to additional meetings and publications, and to new educational initiatives at other universities and institutions.

## LOGISTICS and INTENDED AUDIENCE

This will be a 3-hour workshop, with a target attendance of 20 participants. We hope to attract participation from both academia and industry, and much of the workshop will be devoted to understanding the specific needs of industry with respect to the computer science fundamentals that software engineers should understand and be able to apply.

## ACKNOWLEDGMENTS

The development of the Computational Thinking for Practicing Engineers course is supported by NSF under grant CNS0836133.

## REFERENCES

- [1] D. Garlan, D. Gluch and J. Tomayko. "Agents of Change: Educating Software Engineering Leaders." In *IEEE Computer*, Vol. 30/No. 11, pages 59-65, 1997.
- [2] J. Wing. "Computational Thinking." *Communications of the ACM*. Vol. 49/No. 3, pages 33-35, March 2006.