Teaching Debugging Collaboratively—Midway Report

Sammy Furr

May 7, 2020

1 Introduction

Debugging is invaluable in writing and understanding code, yet it is rarely formally taught. We typically teach students programming structures, concepts, and languages, but we leave them to learn the tools they use to code by themselves. This approach often works well—the programmer's choice of editor is *very* personal, students figure out how to configure an individualized workflow. Perhaps because debuggers are tools, they often get lumped into the "teach yourself" category. Unlike editors or reference guides however, effectively using a debugger requires a set of high-level, platform agnostic, teachable skills. Teaching these skills is effective, and translates into better, faster, debugging and programming. [1] [2]

Teaching debugging is particularly important in

2 The Value of Teaching Debugging

3 Debugger: Low Level

4 Debugger: High Level

5 Next Steps

References

[1] Li, C., Chan, E., Denny, P., Luxton-Reilly, A., and Tempero, E. Towards a framework for teaching debugging. In *Proceedings of the Twenty-First Australasian Computing Education Conference* (New York, NY, USA, 2019), ACE 19, Association for Computing Machinery, p. 7986.

[2] Michaeli, T., and Romeike, R. Improving debugging skills in the classroom: The effects of teaching a systematic debugging process. In *Proceedings of the 14th Workshop in Primary and Secondary Computing Education* (New York, NY, USA, 2019), WiPSCE19, Association for Computing Machinery.