Subgoals for CS1 in Python

Briana B. Morrison
bbmorrison@virginia.edu
University of Virginia
Charlottesville, Virginia, USA

Adrienne Decker
adrienne@buffalo.edu
University at Buffalo
Buffalo, New York, USA

Lauren Margulieux
lmargulieux@gsu.edu
Georgia State University
Atlanta, Georgia, USA

Austin Cory Bart
acbart@udel.edu
University of Delaware
Newark, Delaware, USA

ABSTRACT

In our previous research we found that teaching novice programmers introductory programming in Java using subgoal labels led to deeper knowledge [2] and increased persistence for students potentially at risk of dropping out or failing their first undergraduate course in CS [3]. Subgoals are an instructional tool that is designed to bridge the gap between novices and experts, i.e., students and instructors. Experts often have difficulty explaining concepts at a level that novices understand because they have automatized much low-level knowledge. The task analysis used to identify subgoals makes this knowledge explicit. Subgoals are often expressed to students through subgoal-labeled worked examples that explicitly state the conceptual knowledge expressed through examples. This instructional design of examples allows students to see past superficial details of the example to the structural problem-solving procedure being exemplified [3].

Given the increasing number of universities using Python in introductory programming classes, we have begun the process of defining the subgoals to use in Python-based introductory courses. We have repeated the Task Analysis by Problem Solving (TAPS) development process [1] that we used to create Java-based instructional materials. As with the Java-based instructional materials, we have created subgoals for both evaluating programs written in Python and writing programs in Python. We have identified subgoals for evaluating expressions, conditionals, loops, and functions. For writing programs, we have identified subgoals for writing expressions, conditionals, loops, functions, and using libraries. We are seeking feedback from the community on the identified subgoals before developing the instructional materials, including worked examples and practice problems. We are also seeking feedback and input on the proposed research experiment to assess the impact of the instructional materials.

The poster will present an overview of the TAPS development process for identifying the Python subgoals and the proposed experimental design for assessing the effectiveness of the instructional materials. The original study used a quasi-experimental design. Two sections of an introductory programming course were taught in Java used subgoal-labeled worked examples during class along with subgoal-labeled practice problems while the other sections of the course used traditional practice problems during class. Students in all sections completed the same quizzes, lab and programming assignments, and exams. We analyzed "Explain in Plain English" quiz questions [2], overall student performance on quizzes and exams [3], and self-reported survey answers. We anticipate a similar experimental design for evaluating the Python based subgoal instructional materials with the exception of using the "Explain in Plain English" quiz questions due to the labor and time required for analysis of such responses. We intend to have a similar quasi-experimental situation with two sections of the introductory class learning through subgoal-labeled worked examples and the remaining sections of the class taught in the traditional format. We will again have quizzes and exams to assess student performance. We hope the CS Education research community will provide feedback and options for improving our experimental design, including a replacement for the Explain in Plain English quiz questions.

Both Java and Python subgoals will be available along with information on how to use the instructional materials in the classroom. All instructional materials are freely available from the project website, https://www.cs1subgoals.org/. Instructional materials are implemented in an e-book available at Runestone Academy.

CCS CONCEPTS

• **Social and professional topics → K-12 education**; **Computer science education**.

KEYWORDS

subgoals, introductory programming, Python

ACM Reference Format:
Briana B. Morrison, Adrienne Decker, Lauren Margulieux, and Austin Cory Bart. 2022. Subgoals for CS1 in Python. In *Proceedings of the 2022 ACM Conference on International Computing Education Research V.2 (ICER 2022), August 7–11, 2022, Lugano and Virtual Event, Switzerland*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3501709.3544283

ACKNOWLEDGMENTS

This work is funded by NSF Grants #2143069 and #2110156. We would especially like to thank Nicki Choquette for her help during our meeting and the wonderful maintenance staff in Rice Hall at UVA for retrieving lost papers.

ICER 2022, August 7–11, 2022, Lugano and Virtual Event, Switzerland
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9195-5/22/08.
https://doi.org/10.1145/3501709.3544283

REFERENCES

[1] Richard Catrambone. 2011. Task analysis by problem solving (TAPS): Uncovering expert knowledge to develop high-quality instructional materials and training. In *Learning and Technology Symposium, Columbus, GA.*

[2] Adrienne Decker, Lauren E Margulieux, and Briana B Morrison. 2019. Using the SOLO taxonomy to understand subgoal labels effect in CS1. In *Proceedings of the 2019 ACM Conference on International Computing Education Research.* 209–217.

[3] Lauren E Margulieux, Briana B Morrison, and Adrienne Decker. 2020. Reducing withdrawal and failure rates in introductory programming with subgoal labeled worked examples. *International Journal of STEM Education* 7, 1 (2020), 1–16.