

## Why do students make mistakes? An Antipattern Documentation about Misconceptions related to CS1 Introductory Programming Courses in Python

Guilherme Gama\*, Ricardo Caceffo, Renan Souza, Raysa Benatti, Tales Aparecida, Tania Caldas, Islene Garcia, Rodolfo Azevedo

### Abstract

The purpose of this research is to identify the misconceptions held by undergraduate students when taking introductory CS1 courses using Python. The methodology of this work consisted of interviews with instructors of previous sections of an introductory CS1 course in Python at Unicamp, and through the analysis of past exams. As a result of this work, we documented a set of 28 hypothetical misconceptions in Python through the antipattern [1] format, allowing the identification of why, how, and where the mapped misconceptions usually occur. Future work involves the development of a Concept Inventory—a multiple-choice questionnaire in which each misconception is mapped to an incorrect option—in the Python programming language.

### Key words:

Misconception, Introductory Programming, Concept Inventory.

### Introduction

A Concept Inventory (CI) is a set of multiple-choice questions that can be used to assess students' comprehension on some topic at some point during a course [2]. The purpose of this research is to develop a CI for Computer Science Introductory Programming Courses in Python, as part of our larger research effort in developing a CI for Introductory CS in a more general sense. Our adopted methodology includes the analysis of past exams, interviews with instructors of previous sections of Python-based CS1, and partial adaptation of our own existing work in the C language [3,4].

### Results and Discussion

Using our C language CI as a starting point, we have analyzed 204 past exams in Python CS1 and interviewed three instructors from those sections. The result is a list of 28 hypothetical misconceptions in Python, of which 17 were present in our previous work in C, and 11 have been raised in our work in Python.

Current results indicate that the topics of recursion and iteration pose the same challenges for students of C and Python alike. Issues regarding function parameters and variables have not appeared as prominently in our investigation in Python as they had in C, and we have not been able to ascertain whether Boolean expressions are any source of difficulty in Python. Moreover, we have found that there are certain Python-specific matters that might challenge introductory-level students, e.g. the peculiarities of for-loops in Python and the use of classes and their methods and attributes.

### Conclusions

We have found that the processes of learning computer programming at an introductory undergraduate level in Python and C share certain similarities in terms of general programming concepts, and have certain important differences regarding how these concepts are implemented in the structure of distinct programming languages, such as, in this case, C and Python.

Future work in this study includes determining in more specific terms which of our hypothetical misconceptions are verifiable, through interviews with students and the creation of an online questionnaire, in order to provide statistical feedback on our list of 28 hypotheses.

### Acknowledgement

This research is supported by grant #2014/07502-4, São Paulo Research Foundation (FAPESP). Additional support was provided by the Brazilian Federal Agency for Support and Evaluation of Graduate Education (CAPES), the National Counsel of Technological and Scientific Development (CNPq) and the University of Campinas (Unicamp). We would like to thank Dr. Marco Aurélio Gerosa, Dr. Eduardo Aranha, Dr. Heiko Hornung, Dr. Jacques Wainer, and Dr. Eduardo Xavier for the support and orientation in this research.

<sup>1</sup> EL-ATTAR, M.; MILLER, J (2006). Matching Antipatterns to Improve the Quality of Use Case Models. In *Proceedings of the 14th IEEE International Requirements Engineering Conference (RE '06)*. IEEE Computer Society, Washington, DC, USA, 96-105. DOI=<http://dx.doi.org/10.1109/RE.2006.42>

<sup>2</sup>ALMSTRUM, V.; HENDERSON, P.; HARVEY, V.; HEEREN, C.; MARION, W.; RIEDESEL, C.; SOH, L.; TEW, A. (2006) Concept Inventories in computer science for the topic discrete mathematics. In *Proceedings of the Working group reports on ITiCSE on Innovation and technology in computer science education, ITiCSE-WGR '06*. ACM, New York, NY, USA, 132-145. DOI: <https://doi.org/10.1145/1189215.1189182>

<sup>3</sup>CACEFFO, R.; WOLFMAN, S.; BOOTH, K. 2016. Developing a Computer Science Concept Inventory for Introductory Programming. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16)*. ACM, New York, NY, USA, 364-369. DOI=<http://dx.doi.org/10.1145/2839509.2844559>

<sup>4</sup>CACEFFO, R. FRANÇA, B.; GAMA, G.; BENATTI, R.; APARECIDA, T.; CALDAS, T.; AZEVEDO, R. (2017) An Antipattern Documentation about Misconceptions related to an Introductory Programming Course in C. In *Technical Report 17-15*, Institute of Computing, University of Campinas, SP, Brasil. 42 pages. October, 2017.