Introducing Programming through Lisp

Brian Christian

bpc6565@rit.edu

(585) 475-2995

23 March 2022

20 Lomb Memorial Drive,
Rochester, NY 14623

Table of Contents

Summary

This study strives to evaluate the pedagogical potential of introducing programming to new computer science students through a functional programming approach as opposed to the traditional procedural or object-oriented approach. This is an *Interdisciplinary study* connecting computer science and pedagogy and should be of great interest to computer science academics; if it is found that there is a more accessible approach to teaching computer science to new students, this could increase the quality of and enrollment in computer science programs nationwide. The author and proposed student researcher is pursuing a Computer Science degree at the Rochester Institute of Technology, has significant experience with functional programming in Lisp and the design and implementation of programming languages, and can confidently fulfill the needs of this study.

Introduction

The question to be studied is as follows: is functional programming in a Lisp-based programming language effective in teaching introductory computer science concepts? According to Philip Guo, Python and Java are the most popular languages for introductory computer science courses. Though these are industry-standard languages, they have shortcomings that can cause students to grow frustrated with the language and their code (e.g. off-by-one errors in list indexing, complex class hierarchies, excessive code repetition due to language constraints). On the other hand, functional programming, especially with Lisp, permits programming more flexibly at a higher level and avoids some of the aforementioned issues, making it potentially better suited for introductory programming.

The core methodology is quite simple: design and implement a beginner-friendly Lisp-based programming language, design a four week introductory computer science class using the language, and finally run a small set of incoming computer science students through the course and evaluate their progress.

This study falls under the *Interdisciplinary study* category of the National Institute for Research's Request for Proposals, as it connects computer science with pedagogy by analyzing how the design and understanding of functional programming languages affect the ability for students to understand basic programming principles.

This project stemmed out of the author's passion for functional programming and programming language theory. They have studied these topics in their own time and wish to pursue a larger research project using them. Furthermore, from tutoring other computer science students, they have noticed common mistakes and misconceptions that could be entirely avoided by teaching from a functional-first perspective. This will advance their educational objectives by providing them with practical experience in programming language design and implementation and significant research experience they can carry into writing a graduate thesis.

Methodology

The first step of the research process will revolve around planning the rest of the research process. Specifically, we will design a Lisp language that suits the needs of our research: beginner-friendly and functional, with some benefits from mainstream languages. We will decide what specific features are most important to include and what can be left out for the sake of simplicity. In addition to the programming language, we will design the overall structure of the course to be taught, including a syllabus, plan for how lessons will flow, and what type of homework assignments to assign. Inspiration will be taken from the student researcher's experience in introductory courses, the faculty advisor's and other RIT Computer Science professors' experiences teaching, and online introductory computer science courses.

Following this initial planning, we will implement the interpreter for the Lisp language of our design. It will be able to run complete program files as well as provide an interactive interface through which users can try out different pieces of code and observe their behavior. We will make sure it can be easily accessed on major operating systems and install it on the RIT Computer Science computers for consistent and hassle-free access.

Along with this, we will create specific lesson plans and homework assignments for each week of the course. Topics will include "Values, Variables, Conditions, and Functions", "Recursion", "Data Structures", and "Algorithms" as these are common subjects to discuss in introductory computer science courses.

In order to garner interested students, we will work with RIT's Computer Science Department to advertise the opportunity for incoming computer science students to get a headstart on learning their subjects in an interesting new format by participating in this program as a paid research participant. Participants will be awarded $200 for participating in the research study, and we will recruit 15 participants based on their existing computer science background, with preference given to students new to the field.

Once research participants have signed up and the Lisp interpreter and lesson plans are complete, we will begin teaching the course according to those plans. There will be one subject per week, for a total of four weeks. We will continuously monitor how students are progressing, modify lessons as needed for their benefit, and evaluate their homework in a timely manner to understand how they are understanding the material and give them useful feedback.

Finally, when the course is complete, we will evaluate the research study overall and prepare a monograph summarizing our findings.

Materials

Material	Purpose
RIT Computer Science Department	- Advertising to potential research participants - Ensuring research participant access to programming tools - Soliciting course advice from professors
Georgia Tech edX Professional Certificate "Introduction to Object-Oriented Programming with Java"	- Study structure of introductory object-oriented programming computer science courses
Georgia Tech edX Professional Certificate "Introduction to Python Programming"	- Study structure of introductory procedural programming computer science courses

Schedule of Work

Week	Description of Work
1	- Meet with faculty advisor to establish research plan - Design custom Lisp dialect for course - Design course structure, including syllabus
2	- Implement Lisp Scanner and Parser - Design Values, Variables, Conditionals, and Functions Teaching Materials
3	- Implement Lisp Evaluator - Design Recursion Teaching Materials - Advertise course/paid research opportunity
4	- Implement Lisp Type System - Design Data Structures Teaching Materials

Week	Description of Work
5	- Implement Lisp Standard Library - Design Algorithms Teaching Materials
6	- Teaching Values, Variables, Conditionals, and Functions
7	- Teaching Recursion
8	- Teaching Data Structures
9	- Teaching Algorithms
10	- Evaluating study results and completing monograph

Budget

Item	Unit Cost	Unit Count	Total
Labor Hour	$20 per hour	400 hours	$8000
Participation Reward	$200 per reward	15 rewards	$4000

Biographical Background

Faculty Advisor

Matthew Fluet is an Associate Professor in the Department of Computer Science at the Rochester Institute of Technology. His research interests include functional programming, programming languages, and compiler construction. As a professor with these focuses, he is well-equipped to provide guidance on introducing new concepts to students, especially functional programming, and for designing a programming language and compiler. Selected publications are listed below and his full Curriculum Vitae can be found here.

- Fluet M. et al. (2010) Programming in Manticore, a Heterogenous Parallel Functional Language. In: Horváth Z., Plasmeijer R., Zsók V. (eds) Central European Functional Programming School. CEFP 2009. Lecture Notes in Computer Science, vol 6299. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-17685-2_4
- Jim Allen, Zena Ariola, Pierre-Louis Curien, Matthew Fluet, Jeff Foster, Dan Grossman, Robert Harper, Hugo Herbelin, Yannis Smaragdakis, David Walker, and Steve Zdancewic. 2010. An overview of the Oregon programming languages summer school. SIGPLAN Not. 44, 11 (November 2009), 1–3.https://doi.org/10.1145/1816027.1816029

Student Researcher

Brian Christian is a third-year Computer Science B.S. student at the Rochester Institute of Technology expecting to graduate in May 2024. Through taking Computer Science for AP Students and other introductory courses and working as a computer science tutor, he understands the structure of introductory courses, how frustrating some concepts can be, and how to clearly explain them to other students. The introductory courses provided him with a strong understanding of Python and Java and their respective strengths and shortcomings and thus how Lisp could be better suited to introductory programming. His intellectual interests include programming language design, functional programming, and Lisp programming languages. As such, several of his independent programming projects relate to these topics: *mal* is an implementation of a Lisp programming language, while *cljox* is an implementation of the Lox programming language. Both are written in Clojure, a popular Lisp dialect. He has also worked on professional Clojure projects for Pyze Inc., a California based analytics startup.

Conclusion

The National Institute for Research should fund this research for two reasons. For one, it has the potential to massively benefit educators and students nationwide. If educators help students understand computer science with more ease, it is likely that more students will complete their computer science degrees, leading to a larger computer science workforce and reducing the stereotype of the field as a supremely difficult discipline. Additionally, there is much potential for industry code to be less bug and error prone by using more functional programming languages and paradigms, saving time and money spent debugging and cleaning up messes after catastrophes. Overall, functional programming has the power to transform education and industry in many ways that could be kickstarted from this very research project.

Work Cited

Guo, Philip. “Python Is Now the Most Popular Introductory Teaching Language at Top U.S. Universities.” *Communications of the ACM*, Association for Computing Machinery July 2014, https://cacm.acm.org/blogs/blog-cacm/176450-python-is-now-the-most-popular-introductory-teaching-language-at-top-us-universities/fulltext. Accessed 22 March 2022.