

# Homework 12 - Introduction and Methods

Andrew Carter and Beryl Egerter

April 23, 2013

## 1 Abstract

We performed a qualitative analysis on a small study asking questions about code evaluation and questions about debugging. The code samples included programming concepts or syntax unfamiliar to the students. We saw different strategies such as following order of execution or reading code from the top of the file down. Strategies often changed from question to question, and certain appeared more suited to different questions. We believe evaluating these strategies is important because a programmer's choice of strategy can affect their success or failure in maintaining, changing, or interacting with code written by others, which often occurs.

## 2 Introduction

Programmers often have to look at code they are unfamiliar with for a variety of reasons whether they are students or professionals. In industry, programmers often work with a large code base and fixing a bug or finding code that does something in particular may not be easy or entirely understandable. Computer science students sometimes start programming assignments with a code base that they are expected to alter or add to, but not necessarily have the background to understand some of it, and may be extremely confused if they cannot figure out where to change or add code. Programmers with any amount of experience need effective strategies to understand existing code depending on the code and their objective.

In our qualitative study we asked two questions about code evaluation and two questions about code debugging. In each of the questions we embedded a concept we did not expect our participants to be familiar with or code that we thought would be hard to understand. These included overly complicated functions, an unfamiliar programming language, or the functional programming concept of passing a function as an argument. Having questions about code evaluation and debugging answered by each participant was important to us since the research we have read has looked at code evaluation on its own or debugging on its own but not together with the same participant. We wanted to know if strategies changed or were more or less effective depending on the situation.

We found that a participant's choice of strategy could affect their success in solving the problem by being able to avoid or understand complicated code or unfamiliar concepts. One participant in particular chose two different strategies for an evaluation problem and a debugging problem. The strategies helped him understand an unfamiliar concept in the evaluation problem and avoid complicated functions in the debugging problem.

The primary contribution of this work is a qualitative analysis of how different strategies helped a student overcome unknown concepts both in code evaluation and in debugging. As a small qualitative study, there are many questions that could be focused on next. In particular, we are interested in other strategies that could be used for each problem, their effectiveness, and what leads students to choose which strategy. We could also follow up on some factors that may have affected some of the decisions students made about strategy in this study such as length of code or access to resources such as Google.

### 3 Data Collection Methods

In this study we did four interviews where two interviewers interacted with a single student. Four students were interviewed in total. For each interview, a student was given four different problems. Each interview took between 40 minutes and 1 hour to complete. After they felt comfortable with the answer to their problem, the interviewers asked the student questions about the problem to gather more information on how they solved the problem, and how thoroughly they understood various sections of the provided code. For each interview, the student’s actions and voice were recorded, and then transcribed and analyzed afterward.

Though the participants all had taken basic programming courses, there were a variety of backgrounds. Despite the varied experience levels, none of the students had completed the programming languages class, where topics such as functional programming are covered in depth. However, all of the students had been exposed to Python syntax and list, as well as the map function in Python. In this paper we focused on one participant who also had tutoring experience for a course in program development and data structures.

The four problems were split up into two evaluation problems and two debugging problems. For the evaluation problems, the student was instructed to evaluate the Python code given to them. For the debugging problems, the student was informed of the unwanted (buggy) behavior, although for the first debugging problem it was asked whether the behavior occurred (which it did), and for the second the student was told that the behavior occurred and was asked to find a solution. The students were encouraged to talk out loud and informed that they could write on the paper. Finally, the interviewers answered questions that could be reasonably inferred from a google search, such as, “is `__init__` a constructor?” However, questions specific to the code were not answered, and when students felt confident in their solution we accepted it independent of correctness. If it became clear that a student was not making progress, hints were supplied, however this did not happen in any of the interviews we analyzed.

## 4 Data Analysis Methods

To analyze the data we made a content log for each interview. The content log was a written record of when the student started answering each question, and timestamps of significant events so that we could easily find interesting portions of the video. We also kept records of what we thought was interesting about the interview.

When planning the interview questions, we had already thought about the strategies students could use to solve the questions. For instance, Question 1 has the execution order, numbers of the function names, and layout of the functions on the page all in different orders so we can differentiate between those strategies. We noticed students switching strategies between problems in our interviews, which we then decided to analyze.

We transcribed several sections, which was too much data to present in this paper. The sections of transcripts presented in this paper best illustrate our findings. In particular, the student we present most clearly shows his confusion, and resolves it in the most clear manner.