

# Homework 05 - Interviews and Content Logs

Andrew Carter and Beryl Egerter

February 27, 2013

## 1 Interview: ACBE3

### 1.1 Post Interview Reflection (Joint)

- Syntax error - immediately said was never on the line the error gives
- Should have asked people about their previous experiences such as tutoring, which could make a huge difference in looking at code
- Huge variety of skill levels we have been getting
- Much better at finding the problem without understanding everything else than previous participants
- Got both of the bugs in Ruby and continued, but stopped immediately after finding bug in python
- debugging problems, resorted to top down (except when given an error with line number)
- for the first two, really didn't need to write on paper to keep track of what was going on
- was thrown off by list syntax in python where code enclosed in brackets means different things in different places

### 1.2 Interview Summary

Interview length: 53:36m

No technical difficulties encountered. We went through Questions 1.1, 1.2, 2.1, and 2.2, and 3.1 (3.1 cut off by time constraints).

### 1.3 Most Interesting Pieces

- Time:  
Summary:  
Relations to Topic:  
Possible Arguments:

- Time:  
Summary:  
Relations to Topic:  
Possible Arguments:

## **1.4 Content Log**

Content log is provided in separate text file labeled ACBE3.

## 2 Interview: ACBE4

### 2.1 Post Interview Reflection (Joint)

- Interesting that changed methods between 1.1 and 1.2
- also, differences between starting point in 1.1 and 1.2 vs. what large clues existed (main function v. not main function)
- If we had switched order of first and second would it have affected things?
- Very verbose
- Much better understanding of functions in 1.1 than in 1.2
- Assumptions that previous coders were intelligent
- Ideas of basic functions, or 'important' functions

### 2.2 Interview Summary

Interview length: 47:47

Technical difficulties:

- No color formatting on paper sheets. Didn't appear to be a serious problem, grayscale affect was visible to differentiate keywords so it wasn't a total loss of information.
- Time constraints meant we needed to hurry/not ask many follow up questions in Question 2.2 - also, participant stuck around briefly after camera was turned off, even though had given us a hard time constraint, which meant that part of the answer to the last question is not on tape.

We went through Questions 1.1, 1.2, 2.1, and 2.2.

### 2.3 Most Interesting Pieces

- Time: 14:40 - 15:40  
Summary: Participant describes differences between Questions 1.1 and 1.2. Mentions that function calls branch more in 1.1. Also mentions that 1.1 has a main function that was a logical start of execution, but in 1.2, had to scan through code to find where statements first occur instead of function definitions.  
Relations to Topic: Gives a sense of what a programmer may first look for in code to find a place to start.  
Possible Arguments: While a programmer can just start from top going down, analyzing code, an efficient approach may be identifying the thread of execution by clues found in function names or syntax.

- Time: 22:50 - 23:48  
Summary: In Question 2.1, participant identifies `is_leap_year?` as a function to pay attention to. Then, by looking through all the code and seeing that the function is not called anywhere, participant claims that leap year support is not implemented. Participant did recognize that it was possible that actual code implemented without the function.  
Relations to Topic: Shows how a programmer can pay attention to just top level constructs (function names and how functions interact) than the actual code computations.  
Possible Arguments: When debugging specific problems in unfamiliar code, keywords in function names and function calls are an efficient strategy.

## 2.4 Content Log

Content log is provided in separate text file labeled ACBE4.

**3 Reflect on Expected Focus**

**4 New Focus?**