COMP61511 (2017) (http://studentnet.cs.manchester.ac.uk/pgt/COMP61511/): Labs

An Interview Like Situation

Job interviews for software developers have become more and more elaborate with Google's interviewing procedure become quite notorious. The key point of and interview is to try to determine whether you have *specific* capabilities. Those capabiltiies might be *suggested* by your resumé, but just because you passed a programming course doesn't directly tell us you can program! Hence the rise of little programming exercises.

In this lab, you will undergo an hour long analogue of a programming interview. The goal of this lab is to get us all up to speed with some simple tasks and give an experiential ground for the afternoon lecture. You will fill in a simple background questionaire and complete two simple programming tasks.

Prerequisites

You should be competent with:

1. Blackboard (https://online.manchester.ac.uk/)
2. A command line shell (`bash` , preferably)
3. Python (version 3.x) programming (nothing fancy, basic constructs!)
4. The basic Python infrastructure (e.g., running Python programs; on lab Linux machines, the command is `python3`)
5. The Python Documentation (https://docs.python.org/3.3/)!

Topics

1. Background questionnaire in Blackboard (<10 minutes)
    - Log into Blackboard (https://online.manchester.ac.uk/)
    - Once you are logged in, you can follow this direct link to Lab 1 Questionnaire (https://online.manchester.ac.uk/webapps/assessment/take/launchAssessment.jsp?course_id=_48187_1&content_id=_5693031_1&mode=cpview&target=blank)
    - You have 15 minutes to complete it. But it's dead easy…just questions about your programming background. There's no right or wrong.
2. The FizzBuzz Question (fizzbuzz.html) (<20 minutes)
3. The Rainfall Problem (rainfall.html) (≈30 minutes)

Submission details

1. The Blackboard quiz should be done in Blackboard. It is *timed*!
2. Download lab_1_stub.zip (lab_1_stub.zip) which is an archive with stubs for each of the programs you must write!
3. The archive also includes a `prepare_lab1_submission.py` script. If you run this in the unzipped directory, it will do some basic sanity checking and generate an archive suitable for submission. The invocation is `python3 prepare_lab1_submission.py yourusername` . So for Bijan, it's `python3 prepare_lab1_submission.py mbassbp2`

Collaboration and aid

Your submission must be entirely your own work. And you should only discuss things with an instructor. You are not graded for correctness as this is a *formative* assessment. The point is what you learn, not how well you do!

If you finish quickly

Note that these times are intended to be generous to accomodate people experiencing Python or Linux for the first time or other lab issues. If you finish quickly, get a start on this reading.

- On fizzbuzz (mostly blog posts):
    - Using FizzBuzz to Find Developers who Grok Coding (https://imranontech.com/2007/01/24/using-fizzbuzz-to-find-developers-who-grok-coding/) (the *locus classicus*)
    - Why Can't Programmers.. Program? (https://blog.codinghorror.com/why-cant-programmers-program/) (probably the key populiser)
    - Perfect Example (https://web-beta.archive.org/web/20070303100237/http://burningbird.net/technology/perfect-example/) (a critique of the FizzBuzz interpretation)
    - FizzBuzz In Too Much Detail (https://www.tomdalling.com/blog/software-design/fizzbuzz-in-too-much-detail/) (how to elaborate fizzbuzz to an enterprise program)
- On the rainfall problem (mostly research papers:
    - Learning to program = learning to construct mechanisms and explanations (http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=EB5915FF0D24EBCE1F2CCF49DE86D481?doi=10.1.1.11.6583&rep=rep1&type=pdf) (the seminal paper, may feel a bit old fashioned)

- The Recurring Rainfall Problem (https://pdfs.semanticscholar.org/f772/087a1ef8f524cc2414c3b64636dd0b9985eb.pdf) (nice, recent experiment with a good overview; good starter paper)
- Do We Know How Difficult the Rainfall Problem is? (https://dl.acm.org/citation.cfm?id=2828963) (conatins a detailed literature review as well as study showing high success rates)