**Exercise for IM Developer Role – Solo Capital Partners**

**Sourcing and structuring data from flat files**

The 13F-HR filings represent security holding information that institutional investment managers must file with the SEC on a quarterly basis. For more information refer to the EDGAR website; in particular: <http://www.sec.gov/answers/form13f.htm>

*Complete submission text files* contain the quarterly holdings and other related information for a specific fund manager (identified by their CIK). Your task is to load and structure four abridged *complete submission text files* (which have been provided) so that 13F-HR filing information can be retrieved. The files provided in this exercise specify the quarterly holdings of GSA Capital Partners LLP (CIK 001362033) for the 4 quarters to 12/31/2012. They are indexed by the **period of report** field (the date the holdings relate to) and **filed as of date** field(the date the holding were filed with the SEC).

1. Store the fields from the *complete submission text files* (including Name of Issuer, Title of Class, CUSIP, etc) in a data structure. Be sure to include the “period of report” and the “filed as of date” as fields.
2. Retrieve the following information (for all of these tasks assume common stock is classified as “COM” in the Title of Class field):
3. What was the total value of all common stock positions in the fund for each quarter? Did the fund grow or fall in value with respect to its common stock positions over the 4 quarters?
4. What would have been the 5 largest holdings of common stock that were publically available on 12 August 2012 for the fund manager?
5. As at 12/31/2012, what were the fund’s 3 biggest new common stock positions (stocks it had not held in the previous quarter)?

Technical constraints (please contact [matt.savage@solo.com](mailto:matt.savage@solo.com) for clarification or to request an exception):

* Please submit your solution as a git, mercurial or subversion repository (or another vcs, by agreement) with frequent commits. We are interested in seeing how you work up to a solution, as well as the final submission. We understand that this might mean that you commit broken, unfinished or exploratory code that you would perhaps not normally push. That’s fine.
* Please also include any tools you might build for yourself during development.
* The basic acceptable solution is a command-line program that extracts the data and prints the answers to the questions. Running the program can optionally require the execution of a build/deploy script. If you find that adding some kind of UI is helpful to you, submitting the solution as a UI is fine as well, although there should be a very simple mechanism to retrieve the answers to the questions above.
* Please include full instructions for executing the programme, including running a build.
* Preferably the solution should run on Windows 7, although OSX 10.8.2 is also acceptable. Requiring Ubuntu is possible if you have really compelling reasons to use Linux (providing a Vagrant file and puppet script/chef recipe would be appreciated, but isn’t necessary).
* You have some flexibility to choose your language/platform. Using one or more of these technologies is acceptable - C#, VB.Net, F#, Python, Ruby, Scala, Clojure, Java, JavaScript, shell scripts. Other languages are possible, by agreement.
* The solution can only make use of the working directory and temporary files on the host machine. Deleting the working directory and any temporary files should remove all trace of the programme having been built or executed (please be considerate of your code reviewer’s machine).
* You can assume that the host system has all of these technologies installed (where relevant)- OSX 10.8.2, Windows 7, JDK 7, .Net 4.5, Python 2.7.4 & 3.3.1, Ruby 2.0.0 (and rvm/pik), Python (and virtualenv/pip), Windows Powershell. Everything else should be bundled with your solution.
* Within the constraints above, the solution can use any external dependencies you feel are appropriate (either bundled or downloaded as part of the build using nugget/gems/maven/etc.). In particular, please note the constraint that the solution can only access the working directory and temporary directories. So if the solution is submitted in Ruby, for example, please include automated scripts calling out to RVM or similar rather than install gems globally. If you’re using .Net, please don’t put anything in the GAC.