LAUNCH SCHOOL PREPARATION

This document will be a work in progress, as the curriculum will be planned one month at a time, based on how far each person gets and a consensus on what should be studied next.

Some background information. Launch School emphasizes the fundamentals of Web Development: Object Oriented Programming, Problem Solving Skills, HTML, CSS, HTTP, and SQL. The big difference in Launch School's approach is the emphasis on "Mastery based learning". Instead of just giving students an introduction to different technologies, those essential basics which do not change are learned to some level of mastery. A break is made after course 249, where some students are invited to participate in the Capstone program while others can choose to move onto the Frameworks and Integrations module.

The assessments at Launch School are extremely demanding. You must know the material very, very well to pass.

My objective in this preparatory curriculum is to not only to help prepare you for the demands of Launch School but also to give you a understanding of how much time outside of work you need to commit to make adequate progress with the program.

Phase 1: Complete the Launch School Preparatory Work by the end of September.
Additional work required for Phase 1:

- I have assigned additional readings from ruby-doc.org for each chapter of the Launch School Book "Introduction To Programming With Ruby"
  - For The Basics Chapter
    - Read the Literals Rdoc from the Ruby Documentation http://ruby-doc.org/core-2.4.1/doc/syntax/literals_rdoc.html
    - Read the Precedence Rdoc from the Ruby Documentation http://ruby-doc.org/core-2.4.1/doc/syntax/precedence_rdoc.html
    - Read the Minitest documentation (http://docs.seattlerb.org/minitest/) and write a few of your own tests to make sure you can run the tool
    - Write one or more tests based on topics from the Literals or Precedence documentation or "The Basics" chapter. I will only mention this once, but I strongly advocate writing occasional tests for topics you are learning. Minitest is the simpler of the two main testing libraries used by Rubyists.
  - For the Variables Chapter
    - Read the Assignment Rdoc from the Ruby Documentation http://ruby-doc.org/core-2.4.1/doc/syntax/assignment_rdoc.html
  - For the Methods Chapter
    - Read the Methods Rdoc from the Ruby Documentation http://ruby-doc.org/core-2.4.1/doc/syntax/methods_rdoc.html
    - Read the Calling Methods Rdoc from the Ruby Documentation http://ruby-doc.org/core-2.4.1/doc/syntax/calling_methods_rdoc.html
  - For the Flow Control and Loops and Iterators Chapters
    - Read the Control Expressions Rdoc from the Ruby Documentation http://ruby-doc.org/core-2.4.1/doc/syntax/control_expressions_rdoc.html

LAUNCH SCHOOL PREPARATION

- For the Arrays Chapter
  - Read the Array documentation from the Ruby Rdoc http://ruby-doc.org/core-2.4.1/Array.html, read the introductory part down to but not including the section titled "Public Class Methods". We will pick up class and instance methods later in the course.
- For the Hashes Chapter
  - Read the Hash documentation from Ruby Rdoc down to but not including the "Public Class Methods" section. We will cover class and instance methods later in the course.
- For the Files Chapter
  - Though unrelated, read the Modules and Classes section from Ruby Rdoc http://ruby-doc.org/core-2.4.1/doc/syntax/modules_and_classes_rdoc.htm
- For the More Stuff Chapter
  - Read the Introductory material on regular expressions from the ruby doc http://ruby-doc.org/core-2.4.1/doc/regexp_rdoc.html
- I assume each of you has a GitHub account. Use the Jekyll static website generator to create a blog and post one technical article on either something near and dear to you. Launch School advocates regularly publishing to a technical blog
- Sign up for exercism.io (http://exercism.io/) and do the first 10 Ruby exercisms (not including the first “Hello World” exercism).