Software Development Foundations Using Ruby on Rails

Table of Contents

Starting Point: 2

Goal: 2

Daily Format: 3

Instructional Time (approx.): 3

Lab Time: 3

Curriculum: 4

Unit 1: TDD of Ruby Data Structures 4

Capstone: Unix Cal 4

Week 1: Warm-up and Basic Data Structures 4

Week 2: Digging into Cal; Basic sorting and searching algorithms 5

Week 3: 6

Week 4: Finishing Cal; Data storage formats (e.g. json, csv, etc.); 7

Unit 2: Exploring the Ruby Ecosystem and Standard Library 8

Capstone: Pure Ruby TDD Resume Piece 8

Week 5: Gem Ecosystem and Sinatra 9

Week 6: Interacting with external systems (APIs, Databases, Queuing); 10

Unit 3: Simple Rails Applications 11

Capstone: Building a Rails Blog 11

Week 7: rails new <project>; vanilla rails & generators; collaboratively planning the project; rails testing options; 11

Week 8: Live coding the blog 12

Week 9: RSS feeds; exposing an API; securing your application 13

Unit 4: Advanced Rails / Survey of Software Engineering 14

Capstone: TDD “client” project 14

Week 10: Project Refactoring; peer code review; How would you do it differently? 15

Weeks 10+ : Advanced Topics / Guest Lectures 17

Mentor schedule: 19

NBIC Speaker/Directions Summary: 19

# Starting Point:

Our students currently have some skills with HTML/CSS/Javascript and have, to some extent, been exposed to client work and wordpress/php.

The students have a strong focus on the end-result of being professional software developers.

# Goal:

To produce hirable Junior Developers. As such, they should be able to:

* build on existing Rails applications
* work on cross-disciplinary teams
* grow their skills independently
* understand the depth/breadth of the field

# Daily Format:

The daily format is 4 hours of instructional time followed by 4 hours of lab time.

## Instructional Time (approx.):

|  |  |
| --- | --- |
| 75m: | Go over individual tasks and research from the previous day. Students will be expected to spend 4 minutes each discussing the results of their work. |
| 5m: | *Break* |
| 70m: | 30m: Topic 1  10m: Transition Time and Discussion  30m: Topic 2 |
| 15m: | *Break* |
| 70m: | 30m: Topic 3 / Guest Lecturer  10m: Transition Time and Discussion  30m: Introduce Exercise |

## Lab Time:

During Lab Time, students will work on a variety of exercises and independent research. Teaching assistant(s) will be on hand to assist during lab hours.

# Curriculum:

## Unit 1: TDD of Ruby Data Structures

## Capstone: Unix Cal

### Week 1: Warm-up and Basic Data Structures

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Day 1 | Day 2 | Day 3 | Day 4 | Day 5 |
| Report/  Review | (swapped w/ Topic 1)  Discussing students’ background and experiences | Discussion on breaking a problem down | Review Ruby Koans Progress | Student reports on command line | How did you implement this code? How did you structure it? |
| Topic 1 | - Who am I?  - Course Goals  - Why start with Ruby? | How Objects work in a system; Floating point vs. fixed point numbers | (cont) | (cont) | Exercise: Refactoring a blob of code into classes/methods |
| Topic 2 | The lifecycle of a software project | Exercise: Introducing Ruby’s high-level object types; Introduction to Documentation | Review: Ruby Syntax | (cont) | Data processing (backstory for stacks and queues) |
| Topic 3 | Exploring irb/ruby command line  (esp. syntax) | Exercise: Version control; Forking a project | Review: Math relevant to Software Development; | Exercise: Git – issuing pull requests | Fundamental data types: array, linked list |
| Topic 4 | Introducing Unix Cal; The importance of requirements | Stack traces | Math Exercises | Introduction to Ruby command-line I/O | Review of OO concepts ; UML as a communication tool |
| Exercise | Breaking down a problem into tiny parts | Puzzle through stack traces | Research: Command Line | Command-line I/O (cheers exercise) | Finish Ruby Koans by Monday |

### Week 2: Digging into Cal; Basic sorting and searching algorithms

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Day 6 | Day 7 | Day 8 | Day 9 | Day 10 |
| Report/  Review | N/A | Discuss challenges and implementation techniques |  |  | Ruby Hoedown –  No Class |
| Topic 1 | Fundamental data types: stacks and queues | Catching up on implementation (if needed) | Pair Exercise: Walk each other through your code |  | Ruby Hoedown –  No Class |
| Topic 2 | Further discussion of OO: mixins and more inheritance | How do we search & sort things in real life?  [ Orendorff ] | Fundamental data types: doubly linked lists | Discrimination in Tech | Ruby Hoedown –  No Class |
| Topic 3 | Discussion of TDD: Goals and basic techniques | Exercise: Naïve search  e.g. smallest/largest items in LinkedList (fall back on Array) | Pair Exercise: Doubly Linked Lists | Trees and hashes | Ruby Hoedown –  No Class |
| Topic 4 | Walking through the tests for the exercise | Review exercises (above) and discuss individual assignments | Collaborate on tests | Digging into Cal: How would we describe the functionality? + History of Date/Time Math | Ruby Hoedown –  No Class |
| Exercise | Implementing an array/linked list (with provided tests from the forked repo) | Naïve search; Research/Report on a Search/Sort Algorithm | Pair Exercise:  Naïve sort | Describe how cal works in detail | Ruby Hoedown –  No Class |

### Week 3:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Day 11 | Day 12 | Day 13 | Day 14 | Day 15 |
| Report/  Review | Talking about what we learned at the Hoedown |  | Alternative meeting space – Emma  Vered is out, so we need good notes. | Working on cal (At Emma) | Meet and greet with mentors |
| Topic 1 | Comparators |  | Talking more about how/why to test | Working on cal (At Emma) |  |
| Topic 2 | Magic Numbers | Review hashes; Discuss hashing algorithms (re: sorting, intro: trees) | Reviewing student progress on cal | Working on cal (At Emma) | Big O  (Jeremy has this penciled in) |
| Topic 3 | Going over requirements on Cal project (based on previous day’s requirements elicitation) |  | Working on cal (At Emma) | Working on cal (At Emma) | Working on cal |
| Topic 4 | Start Cal (early) | Start working on Cal | Working on cal (At Emma) | Working on cal (At Emma) | Working on cal |
| Exercise | Start working on Cal | Start working on Cal | Working on cal (At Emma) | Working on cal (At Emma) | Working on cal |

### Week 4: Finishing Cal; Data storage formats (e.g. json, csv, etc.);

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Day 16 / Oct 1 | Day 17 / Oct 2 | Day 18 / Oct 3 | Day 19 / Oct 4 | Day 20 /Oct 5 |
| Report/  Review | What did we learn from cal | No class / Refactoring Cal (based on peer reviews) | Report in on how refactoring went | Let Sam talk for a few minutes about the barcamp roundtable | How did cal go?  What are your resume project ideas? |
| Topic 1 | Review each student’s results | No class / Refactoring Cal (based on peer reviews) |  | What did you learn from yesterday?  What do you wish that you understood better? | Introduce HTTP (rfc2616) |
| Topic 2 | Psuedo code; Documentation | No class / Refactoring Cal (based on peer reviews) | Code Retreat | Going over remaining questions about Ruby. | Further Ruby questions: Splat; blocks vs. procs vs. lambdas; Fixnum |
| Topic 3 | Refactoring; Live refactoring of linked list code | No class / Refactoring Cal (based on peer reviews) | Code Retreat |  | Pick out resume projects |
| Topic 4 | What is/Why Code Review | No class / Refactoring Cal (based on peer reviews) | Code Retreat | Brainstorming what makes a good resume piece | Exercise: Resume project requirements |
| Exercise | Exercise: review a peer’s implementation of the previous exercise | No class / Refactoring Cal (based on peer reviews) | Code Retreat | Propose 3 projects that would make good resume pieces  (and finishing printing a year of cal) | Have a good weekend |

## Unit 2: Exploring the Ruby Ecosystem and Standard Library

## Capstone: Pure Ruby TDD Resume Piece

During this module, the focus will be on producing a small pure-Ruby TDD’d resume piece.

This will require that the student demonstrate mastery of:

* Basic git usage
* Project Documentation (README, comments)
* Writing Unit Tests
* Basic Ruby

### Week 5: Gem Ecosystem and Sinatra

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Day 21 / Oct 8 | Day 22 / Oct 9 | Day 23 / Oct 10 | Day 24 / Oct 11  Meeting at Emma | Day 25 / Oct 12  [Max is out] |
| Report/  Review |  | Discuss impressions from yesterday | (Primarily a work day) | (Primarily a work day) | Standup |
| Topic 1 | Discuss Use Cases; User Types; User Personas | Exercise: Review each other’s stories | (Primarily a work day) | (Primarily a work day) | What is Ruby Gems? |
| Topic 2 | Exercise: User Personas and Features for your project | UML class diagrams | (Primarily a work day) | (Primarily a work day) | Why did I have you install rvm and homebrew? |
| Topic 3 |  | Exercise: Working on UML | (Primarily a work day) | (Primarily a work day) |  |
| Topic 4 | Discuss actual UML notation; Use UML to describe a familiar system (cal?) | Work on projects | (Primarily a work day) | (Primarily a work day) | Work on projects |
| Exercise | Finish writing user stories | Work on projects | (Primarily a work day) | (Primarily a work day) | Work on projects |

### Week 6: Interacting with external systems (APIs, Databases, Queuing);

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Day 26 / Oct 15  [Max out] | Day 27 / Oct 16  Meeting at Emma | Day 28 / Oct 17 | Day 29 / Oct 18  Meeting at Emma | Day 30 / Oct 19 |
| Report/  Review | Standup | Standup | Standup | Standup | Presenting your projects! |
| Topic 1 | Class discussion on manually testing web apps (leading into how to automate) | (Primarily a work day) | More Sinatra | (Primarily a work day) | Presenting your projects! |
| Topic 2 | Introduce Sinatra | (Primarily a work day) | More Sinatra | (Primarily a work day) | Presenting your projects! |
| Topic 3 |  | (Primarily a work day) |  | (Primarily a work day) | Presenting your projects! |
| Topic 4 |  | (Primarily a work day) |  | (Primarily a work day) | Topic? |
| Exercise | Exercise: Use Sinatra to print out a calendar with given month/year  Exercise: Install postgresql | (Primarily a work day) | (Project Work) | (Primarily a work day) | Exercise? |

## Unit 3: Simple Rails Applications

## Capstone: Building a Rails Blog

### Week 7: rails new <project>; vanilla rails & generators; collaboratively planning the project; rails testing options;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Day 31 / Oct 22  Matt out (try to send him a summary email) | Day 32 / Oct 23 | Day 33 / Oct 24 | Day 34 / Oct 25 | Day 35 / Oct 26 |
| Report/  Review | Final Project presentations | Reiterate that books must be purchased | Live Coding Blog | Live Coding Blog | Live Coding Blog |
| Topic 1 | Introduce the 3 books for this section of the course | Planning our Rails blog | Live Coding Blog | Live Coding Blog | Live Coding Blog |
| Topic 2 |  | Imposter Syndrome  Analysis Paralysis | Live Coding Blog | Live Coding Blog | Live Coding Blog |
| Topic 3 |  | Starting a Rails Project | Live Coding Blog | Live Coding Blog | Live Coding Blog |
| Topic 4 | Code Reviews  !!No More While Loops |  | Live Coding Blog | Live Coding Blog | Project Retrospectives (on Summer projects) |
| Exercise | Refactoring  -project summaries  - make your code public | Assignment: Read the first section of Pragmatic Programmer | Assignment: Code comments to request explanations of the code we did in class |  |  |

### Week 8: Live coding the blog

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Day 36 / Oct 29 | Day 37 / Oct 30 | Day 38 / Oct 31 | Day 39 / Nov 1 | Day 40 / Nov 2 |
| Report/  Review | Reading: Pragmatic 1 |  | Reading: Pragmatic2 |  | Reading: Pragmatic 3 |
| Topic 1 | Live Coding Blog | Guest Lecture on Operating Systems | Draw a sitemap on the board | Live Coding Blog | Live Coding Blog |
| Topic 2 | Live Coding Blog | Guest Lecture on Operating Systems | Live Coding Blog | Live Coding Blog | Book Club |
| Topic 3 | Live Coding Blog | Live Coding Blog | Live Coding Blog |  |  |
| Topic 4 | Book Club |  | Book Club |  |  |
| Exercise |  | Exercise: Reply to Comments on Blog | Exercise: Reply to Comments on Blog | Exercise: Display comments properly nested |  |

### Week 9: RSS feeds; exposing an API; securing your application

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Day 41 / Nov 5 | Day 42 / Nov 6 | Day 43 / Nov 7  @ Emma | Day 44 / Nov 8  @ Emma | Day 45 / Nov 9 |
| Report/  Review | Pragmatic 4 | Pragmatic 5 | Meeting at Emma | Meeting at Emma | Pragmatic 6 |
| Topic 1 | Pick Projects |  | Assignment: Add a feature to weeblog | Assignment: Write out the features of your project (no code!!) in long form (as in cukes) |  |
| Topic 2 | Pick Pairs for Features |  |  |  |  |
| Topic 3 |  |  |  |  |  |
| Topic 4 |  |  |  |  |  |
| Exercise |  |  |  |  |  |

## Unit 4: Advanced Rails / Survey of Software Engineering

## Capstone: TDD “client” project

At this point, the class will be focused on accomplishing a “major” Rails project. As such, the expectation will be that they will spend the majority of their lab time working on that project.

#### New Format for Class Time:

Class will now be split into two portions:

1. A variety of guest lectures introducing the depth/breadth of the software engineering field
2. A discussion of more advanced Rails topics, gems and techniques

Where possible, the advanced Rails topics will tie into that day’s guest lecture(s).

### Week 10: Project Refactoring; peer code review; How would you do it differently?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Week 10 | Day 46 / Nov 12  @Emma | Day 47 / Nov 13  @Emma | Day 48 / Nov 14 | Day 49 / Nov 15 | Day 50 / Nov 16 |
| Notes | **Meeting at Emma** | **Meeting at Emma** | Project Progress Presentations | Standup |  |
| Topic 1 | Project Work | Book club:  Pragmatic 7 | Discuss pull requests | Book club: Pragmatic 8 | 9:15/9:30 –DB/Security Best Practices (Ben Stucki, 1 hour) |
| Lunch | Project Work | Emma presentation (lunchtime) | Working on pull request fixes  Project Work | Bytes of Knowledge presentation (lunchtime) | Book club: Passionate 1/2 |
| Topic 2 | Project Work | Project Work | 1:30 – 3:30 Hypermedia APIs (Luke Stokes) | Project Work |  |
| Assignment |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Week 11 | Day 51 / Nov 19 | Day 52 / Nov 20 | Day 53 / Nov 21 | Day 54 / Nov 22  Thanksgiving | Day 55 / Nov 23  Thanksgiving |
| Notes | Keith Marcum - How caching works |  | 8:45 Robert Bauer (Networks/OS?) | No School | No School |
| Topic 1 | Book Club: Passionate 2 | Deploying to Heroku | Project Progress Presentations | No School | No School |
| Lunch | ~~Populr.me Presentation (lunch)~~ | OnLIfe Presentation (lunch) | Moontoast Presentation  (lunch, 11:30) | No School | No School |
| Topic 2 |  | 1PM- Fryman – Dev Ops  Exercise: Deployment | Exercise: Expanding on weeBlog? | No School | No School |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Week 12 | Day 56 / Nov 26 | Day 57 / Nov 27  Kevin AM | Day 58 / Nov 28  Chris Cotton 10AM | Day 59 / Nov 29  Meeting at Emma  Jeremy PM | Day 60 / Nov 30 |
| Notes | Book Club: Passionate 3 & 4 |  | ~~Project Progress Presentations?~~ | Meeting at Emma | Project demos / End of class |
| Topic 1 | Create an interview/job search checklist | (10:00-11:30) Recruiting/Interviewing/Job Search – Chris Spintzyk | Book Club: Passionate 5 | (Wrapping up Projects) | Project demos / End of class |
| Lunch |  | Atiba presentation  (12 or 12:30, no lunch) | CentreSource presentation (lunch?) | Populr.me (lunch provided) – Daniel Nelson | Project demos / End of class |
| Topic 2 | iGo call Monday at 2ish? | 2PM  Fundamental Gates/Circuits (Daniel Nelson) |  | (Wrapping up Projects) | -- |

### Weeks 10+ : Advanced Topics / Guest Lectures

|  |  |
| --- | --- |
| Intermediate/Advanced Rails Topics:   * Queuing (Redis, etc.) * ~~Caching~~ (Keith Marcum) * ~~Deployment~~ (James Fryman/ Eliza) * Integrating with APIs (Matt Mueller) * RESTful APIs (Matt Mueller) * HAML/SASS * Coffeescript * An overview of Rails 2.3.x (in case the students move into legacy code bases)   Topics specifically requested by students:   * Mobile applications with Rails * ~~Server administration and automation~~ * ~~More about command-line git~~ * ~~API development (Luke Stokes)~~ * ~~DB/Security best practices (Ben Stucki)~~ * ~~Networks~~ * ~~What is Dev Ops? (James Fryman)~~ * HAML/SASS * Coffeescript * Vim basics * Learning other programming languages * How other mid-level developers journeyed through being junior developers * Startup experiences * Common junior dev mistakes * How to avoid analysis paralysis * Team management * What are compilers and interpreters? How do they work? (Jason Orendorff?) | Topics on my personal list of "things developers should know":   * ~~Fundamental gates/circuits (w/ Assembly?)~~ (Daniel Nelson) * The mythical man month * Software maintenance * How to build the data structures you want out of arrays and hashes * How to contribute to open source * Functional programming * Waterfall (from a successful practitioner) * Scrum (from a successful practitioner) * Backups and emergency management * How to get paid what you're worth * The morals and ethics of software development * The iron triangle * "The right tool for the job" * ~~How the DNS system works~~ * How to read an academic paper * Basics of Artificial Intelligence * Basics of UX/HCI * The importance of design (both visual and architectural) * Design Patterns (either specifics or an overview) * Interviewing * Working on legacy projects * A walkthrough of real \*production\* Rails apps |

### Mentor schedule:

https://docs.google.com/a/elizabrocksoftware.com/spreadsheet/ccc?key=0AgrAZG6H-NbIdHVxb3I5NXJuUnJfQlJNTFNMMjZLZHc#gid=0

### NBIC Speaker/Directions Summary:

\_\_name\_\_,

My notes say that you'll be presenting on \_\_topic\_\_ \_\_day\_\_ between \_\_time\_\_ and \_\_time\_\_.

We meet in the NBIC training room (instructions below) and we have a whiteboard, chalkboard and projector (1024x768) available for use.

-Eliza

[elizabrocksoftware.com](http://elizabrocksoftware.com/)

[twitter.com/elizabrock](http://twitter.com/elizabrock)

(858) 205-9285

The Nashville Software School is located at the NIBIC (315 10th Ave. North, 37203).  There is free parking in the NBIC lot.  Use the left entrance (it's just to the left of the front door) and we're the first room to the right (98/99).  The door sticks to 98/99 sticks, but I promise it will be unlocked.