PARSEC Ruby Bootcamp
Syllabus & Curriculum
— Lorin Ricker, developer and instructor —

This document presents the Syllabus and Curriculum for the **PARSEC Ruby Bootcamp**, a multi-week professional course in the Ruby programming language, software engineering and development methodologies. The Bootcamp is conducted as a series of classes in three parts:

1) ***Newbie***, a one-week intensive introduction to Ruby and software development, using real-world code and projects, not just textbook “toy” examples;

2) ***Foundational***, a one-week follow-on which builds a complete foundation of Ruby on the basis of the introductory material of the previous week;

3) ***Essentials***, a one-week follow-on which refines and extends material from the Foundational class, focusing on production-quality command-line programs;

4) ***(Meta)Mastery***, a one-week class for experienced developers which emphasizes and explores object-oriented programming techniques, metaprogramming and reflection.

The PARSEC Ruby Bootcamp series is designed to produce (graduate) “World Class Beginners” who, upon successful completion of this coursework, will be: **a**) ready for hire into entry-level technical/team software programming positions throughout business and industry; **b**) ready to participate in Ruby-based software development projects and activities within their own workplace.

The PARSEC Ruby Bootcamp courses are created, produced and conducted by The PARSEC Group, 999 18th Street, Suite 1725, Denver Colorado 80202.

The *purposes and goals* of the Bootcamp are to teach and develop students' core competencies in the following areas:

- Mastery of the Ruby programming language — this is the top-priority goal of these classes.
- Proficient use of prevalent software development tools and environments, including Linux, command-line utilities, Git (repository), and a high-productivity text editor/IDE — this is a high-priority goal.
- Basic and key Software Development skills and methodologies, with emphasis on collaborative, team-oriented approaches and participation — this is also a high-priority goal.
- Orientation to the Ruby on Rails application development framework and other relevant Ruby-based software packages.

These goals will be reached and/or supported by the following *teaching methodologies*:

- Instructor-lead lecture/discussion presentations of key concepts and ideas, with intentional focus on student participation — including student-to-student presentations — throughout.
- Individual and team-oriented participation in exercises, labs and hands-on class projects.
- Preparatory readings and investigations, done by students outside of regular classroom time (*e.g.*, evenings) to optimize the next-day formal presentation of selected material.
- Guest speaker presentations on key contemporary topics relevant to software engineering and product/project development issues, with extensive opportunities for guest speaker-to-student interactions (Q&A, discussions and conversation following presentations).

Class Mechanics: The PARSEC Ruby Bootcamp will be run over four consecutive weeks, with classes convening at 9:00am (MT) Monday through Friday, with a one-hour lunch break, at least one short break during each morning and afternoon session, and concluding at 5:00pm.

The Newbie and Foundational classes are held over two consecutive weeks as a single, cohesive training; students are required to attend the full two-week session as a unit. Although each class cohort will be encouraged to “stay together” for all four levels, students may return to attend the Essentials and/or (Meta)Mastery classes at any time after completing the Newbie and Foundational classes; Essentials must be completed before taking the (Meta)Mastery class.

Bootcamp Course Justification by Career Opportunities —

According to Dice, Inc. president Shravan Goli (as quoted in CIO Magazine Online article "*Open Source Career Opportunities Continue to Abound*", by Sharon Florentine, Wednesday, September 4th, 2013, http://www.cio.com/article/739097/Open_Source_Career_Opportunities_Continue_to_Abound):

> "**The number of available Ruby on Rails positions grew 15 percent year-over-year from 2,307 to 2,654**."

This statistic is based on job posted on Dice.com from August-2012 to August-2013. A similar growth rate is evident for Python programmers, and although the number of Perl positions fell by ~3% and PHP positions remained stable over the same time period, "*...the numbers show a continued demand for open source developers.*"

Goli says that *the open source's community approach to development* is what makes it so appealing, because *it can accelerate an organization's technology development and adoption.*

"Ten years ago, open source was all about Linux and Linux-based technologies," Goli says. "Now, with the cloud, social media, big data and analysis, search and mobile applications all maturing so rapidly, *it makes more sense for companies to leverage the community effort* to accelerate development and deployment."

Colorado ranks among the top-10 states exhibiting the greatest job opportunities (according to CIO Magazine Online's article "*Top 10 States for Tech Job Growth*", by Rich Hein, Tuesday, August 10th, 2013, http://www.cio.com/article/738463/Top_10_States_for_Tech_Jobs_Growth), conclusions which are based on statistics of growth from the Bureau of Labor Statistics Computer Systems Design and Related Services category. The other nine states are: New Jersey, Massachusetts, Missouri, Texas, New York, Washington, Maryland (including Washington DC), Illinois and Pennsylvania.

And while projections continue to show that corporations are soaking up open source programmer talent at increasing levels, there are strong opportunities for those same employees in entrepreneurial start-ups, funded start-ups and established small business enterprises as well.

Some employers note with concern that a great many candidates for Ruby on Rails positions are not particularly well prepared with mastery of Ruby programming language fundamentals.

Thus, the Bootcamp's primary goal is to train and prepare World Class Beginners, with excellent grounding in Ruby programming foundations, who are qualified to gain employment or professional assignments in precisely this marketplace.

Prerequisites (per student) —

A. Student prerequisites for the components of the Bootcamp are as follows:

1. **Newbie** and **Foundational** — Although this two-week class sequence is designed for newcomers to the Ruby programming language and general software development, exposure to college-level computer science coursework and/or on-the-job software development experience will give the student considerable advantage in this class.
2. **Essentials** — Completion of the **Newbie** and **Foundational** two-week class sequence, or equivalent other coursework (at Instructor's and/or PARSEC's discretion and acceptance), or equivalent programming experience.
3. **(Meta)Mastery** — Completion of the **Newbie**, **Foundational** and **Essentials** classes or equivalent other coursework (at Instructor's and/or PARSEC's discretion and acceptance), or equivalent programming experience.
4. *General* — On-the-job and/or coursework experience with at least one (or more) of the following will give the student considerable advantage in these Bootcamp classes: • Text editing • Applications in a "classical" language (*e.g.*, C, Pascal, Fortran, VB, *etc.*) • Debugging • Shell scripting (*e.g.*, bash, DCL, *etc.*) • Command-line utilities • Operating systems fundamentals •
5. This Bootcamp class series is *not appropriate* for absolute beginners in software development.

B. Laptop: Because it is commonly accepted in most professional environments that a software developer will possess and use his or her own personal laptop computer for programming and project activities, including both employment/on-the-job work and community/volunteer projects, the Ruby Bootcamp requires that *each student bring their own laptop computer to each class*. Each student must provide and use his or her own laptop computer throughout the class, configured for: **a**) dedicated/single boot with Ubuntu Linux (preferred); or **b**) for dual-boot with Ubuntu and Windows; or **c**) Mac/OSX; or **d**) a Virtual Machine, hosted on the laptop's native operating system, running Ubuntu Linux as a guest — VirtualBox is recommended, with the VM's guest disk (container file) sized at no less than 60 GB capacity, with NAT networking enabled, and with Guest Additions installed.

Although a student may have an employment-related and/or personal need and/or desire to work within the Mac/OSX or Windows environment, this coursework will focus primarily on software development within the Linux (Ubuntu) environment, with minimal coverage of the specifics of Mac/OSX, and near silence on Windows.

Coursework will be almost entirely Linux (Ubuntu) oriented, so it is strongly recommended that the student's laptop have the latest Ubuntu release (currently v14.04-LTS, "Trusty Tahr") installed either as the sole OS or as a dual-boot option alongside an existing Windows partition. Other Linux distros (non-Ubuntu) are acceptable if cleared with the Bootcamp's Instructor. A Mac computer with OSX is also acceptable. A Windows-only laptop is not acceptable for this Bootcamp.

Each student should arrange to pre-configure his or her own laptop computer (*i.e.*, install Ubuntu, if necessary) prior to arrival for the Introductory class, and should maintain the Ubuntu (and added software packages) configuration for the subsequent Intermediate and Advanced classes. The Bootcamp Instructor can provide written OS-installation and configuration instructions to any student who requests them, provided that such request is made at least fourteen (14) days prior to the start of class.

C. Wireless network access, both to the Internet and among participants' laptops, will be provided during each class.

D. Software: Students will learn sufficient skills about installing and configuring Linux (Ubuntu) and other Free/Libre Open Source Software (FLOSS) packages to be able to configure their own personal laptop or desktop system.

Each laptop's Linux partition (or VM) will be configured as follows:

- Current Ubuntu (Linux) distro, currently 14.04 "Trusty Tahr"
- Default user account `sysadmin` (password `SysAdmin`): will be used to create at least one developer (personal) account per laptop
- Ruby enVironment Manager (RVM) to support Ruby (MRI) versions 1.9.3 and/or 2.x
- Sublime Text editor/IDE (free evaluation copy)

Each student will, over the course of the Bootcamp, install the following components (at a minimum) on their own laptop computer (note that these are all free/open-source packages, available at no cost to the student):

- Linux packages:
  - Sublime Text (free evaluation copy)
  - Git (revision control repository)
  - RVM (Ruby enVironment Manager)
  - Ruby (latest version)
  - (others to be determined)
- Ruby Gems:
  - Rake
  - pry
  - RSpec / RDoc / ri
  - (others to be determined)

Text Editor —

The text editor is perhaps *the programmer's most important and frequently-used software tool*. And although developers who use other languages (e.g., Java, C, Perl, Python, etc.) usually rely on text editors which are embedded in more comprehensive (and costly) "Integrated Development Environments"(IDEs), Ruby's syntax and development environment is considerably more straight-forward, and such IDE resources provide only complexity and confusion without contributing value to the experience...

Fortunately, the Ruby community enjoys (and has largely embraced) a relative newcomer to the text editor toolkit, Sublime Text (http://www.SublimeText.com), which is exquisitely attuned to the Ruby programming language and attendant development cycle. It runs cross-platform (available for Linux, Mac/OSX, and Windows). It is an excellent choice for Ruby Bootcamp students.

And, although Sublime Text is sold for professional use on a single-user licensing basis (its EULA is not a GPL), it is made available to any potential user for unrestricted evaluation, without feature-disabling or functional timeouts. Sublime Text will be used during the Bootcamp for all text editing tasks and requirements, as an exemplar of a modern, advanced-feature editor. And because it is reasonably priced (currently ~$70 for the single-user license), students can elect to purchase Sublime Text for their own personal use after the conclusion of the Bootcamp.

Teamwork —

All students will be required to work together in team units, ranging from paired-programming sessions to four-to-six person development teams working on class projects. Because this participation is viewed as an essential, non-optional experience requiring in-classroom, physical presence, the Ruby Bootcamp does *not* allow or support "distance learning" (e-classroom).

Students will be strongly encouraged to help each other, in true peer support and collaborative learning: "*If you don't know, ask for help. If you really get it, help someone else.*"

Students will be expected to "present" material to each other (the class), ranging in scope from individual units of Ruby material, solutions to exercises, original research, answers to questions, even including full project designs and defending particular implementation approaches.

This approach models "the real world" of software development, where small, agile teams work closely together to produce high-quality software products, and where community support and participation is valuable, recognized and rewarded.

Student Evaluations —

Student evaluation (grading), including recognition for successful completion of the coursework, will be based on the following:

- Attendance, with: **i**) no more than one (1) excused absence within any single class-week; and **ii**) no more than two (2) excused absences over the course of the entire series of Bootcamp classes; and **iii**) with demonstrated and Instructor-accepted diligence and evidence of remedial (catch-up) efforts on any material/content missed due to absence.
- Various formal and/or informal pop quizzes at the discretion of the Instructor.
- Student's vigorous, willing and productive participation in discussions, Q&A, preparatory activities, and especially in his/her team participation and contribution.
- Student's own formal evaluation, conducted and provided by other class (team) members, as to the student's participation, contribution and efficacy as a team/project member.
- The Instructor shall be the ultimate determiner (arbiter) of each student's final evaluation, and the completion of each class of the Bootcamp shall be based on a Pass/No-Pass grade.

Primary Course Textbook —

A copy of this reference book will be provided to each student:

	Publisher, Year, ISBN	*List*	*Used* *
Programming Ruby 1.9 & 2.0 *The Pragmatic Programmers Guide* — Dave Thomas, with Chad Fowler and Andy Hunt	Pragmatic Bookshelf, 2013 978-1-93778-549-9	$50.00	~ $30.00

The "Programming Ruby" book is commonly known as "the Pick-Ax book" in Ruby circles because of the picture/logo on its front cover. This volume, now in its third edition edition, has been a main-stay of Ruby learning and reference since 1999, and will become a comprehensive and often-used resource for students during the Bootcamp and throughout their careers.

Other Textbook Resources —

A substantial library of Ruby and Rails text and reference books has evolved over the past decade, and the Bootcamp will make use of a selection of these (but will not provide or require them as part of the course itself). This list is not comprehensive, merely representative:

	Publisher, Year, ISBN	*List*	*Discount* *
Build Awesome Command-Line Applications in Ruby 2 (2nd Ed.) — David B. Copeland	Pragmatic Bookshelf, 2013 978-1-93778-575-8	$30.00	~ $20.00
The Ruby Programming Language — David Flanagan and Yukihiro Matsumoto	O'Reilly, 2008 978-0-59651-617-8	$39.99	~ $19.00
Ruby Cookbook (2nd Ed.) — Lucas Carlson and Leonard Richardson	O'Reilly, 2014 978-1-44937-372-9	$49.99	~ $34.35
Pro Git (2nd Ed.) — Scott Chacon and Ben Straub	Apress, 2014 978-1-48420-077-3	$59.99	~ $53.99
Mastering Regular Expressions (3rd Ed.) — Jeffery E.F. Friedl	O'Reilly, 2006 978-0-59652-812-6	$49.99	~ $20.99
Regular Expressions Cookbook (2nd Ed.) — Jan Goyvaerts and Steven Levithan	O'Reilly, 2012 978-1-44931-943-4	$49.99	~ $33.34
Ruby on Rails Tutorial (2nd Ed.) — Michael Hartl	Addison-Wesley, 2013 978-0-32183-205-4	$44.99	~ $29.00
Ruby *for* Rails — David A. Black	Manning, 2006 978-1-93239-469-6	$44.99	~ $4.25
The Well-Grounded Rubyist — David A. Black	Manning, 2009 978-1-93398-865-8	$44.99	~ $19.50
Eloquent Ruby — Russ Olsen	Addison-Wesley, 2011 978-0-32158-410-6	$39.99	~ $27.00
Design Patterns in Ruby — Russ Olsen	Addison-Wesley, 2007 978-0-32149-045-2	$54.99	~ $21.50
Metaprogramming Ruby (2nd Ed.) — Paolo Perrotta	Pragmatic Bookshelf, 2014 978-1-94122-212-6	$38.00	~ $26.79
Ruby Best Practices — Gregory Brown	O'Reilly, 2009 978-0-59652-300-8	$34.99	~ $3.50

** Students are encouraged to purchase used professional textbooks whenever feasible. Used books prices varies widely based on book condition, age and seller. Prices listed here are samplings from Amazon.com's used book affiliates at an arbitrary point in time, and are not intended to be anything more than comparative.*

Online Resources —

As for most FLOSS programming languages and environments, there is a wealth of online (web-based) informational resources available for the Ruby community, and the Bootcamp will refer to and make use of as many of these as practical, in an effort to demonstrate and instill informational independence for each student. These include (this is not an exhaustive list):

- http://www.ruby-lang.org/en/ — The official Ruby website
- http://ruby-doc.org/ — The official documentation site for the Ruby language, core API, standard library, and many commonly-used gems
- http://rubygems.org/ — The official RubyGems website
- http://github.com/ — A community site for open source Ruby (and other) projects
- http://rubyonrails.org/ — The official Ruby on Rails website

Weekly/Daily Class Schedule

The next pages provide a topical outline, followed by the week-by-week, day-by-day top-level schedule for the Bootcamp.

Each page outlines the topical contents of each day's morning and afternoon sessions. Each morning and afternoon session will include one or more breaks as needed.

Each session consists of (either exclusively or in some appropriate combination):

a) Lecture (informal style) with full student discussion and feedback;

b) Exercises, guided hands-on efforts where students begin to apply knowledge and approaches just covered in lecture/discussion;

c) Lab activities where, in groups of small teams, students apply and practice learned skills to class projects;

d) Guest-speaker presentations, including a formal presentation, but followed immediately by informal student interaction, including Q&A, discussion, more in-depth conversation and relationship-building.

Notes:

1. Because a certain amount of basic and getting-started material must be covered in order to allow students to commence productive exercises and lab work, Week 1 is weighted with more lecture/discussion hours than subsequent weeks.
2. Students should *not* make return-home travel plans for Friday afternoons, as each class week will include a full in-class day on Friday, much of which will be devoted to students' presentations of team projects.

Ruby Bootcamp Topical Outline

Newbie — Week 1

- What is Ruby, and what is it used for?
- An overview of Ruby
- Laptop resources: Linux, command line, Ruby and RVM, Sublime Text
- Programming teams, pair programming
- Ruby —
    - puts (output)
    - gets (input)
    - strings, string interpolation
    - expression values and objects
    - true/false, conditions
    - statements, expressions
    - flow control, if/then/else, case, loops
    - iterators
    - intro to blocks
    - program structure
    - naming conventions
    - irb (interactive Ruby)
    - intro to object oriented programming
    - objects
    - methods and messages
    - data types
    - invoking methods
    - arguments, parameters and defaults
    - return values
    - duck typing
    - build-in data objects
        - numerics
        - string & regexp
        - array
        - hash
    - enumeration and iteration
    - introduction to git and github
    - Ruby library (intro and survey)

Foundational — Week 2

- Ruby —
    - strings (in more depth)
    - interpolation
    - searching and substitutions
    - string library methods
    - intro to regexps and pattern matching

        - Ruby library (in more depth)
            - string methods
            - numeric methods
            - hash and array methods
            - file methods
            - kernel and process methods
- Ruby Gems (intro)
- building a command line interface (OptionParser)
- configuration files (YAML)
- intro to debugging

Essentials — Week 3

- Ruby review (kick it up a notch)
- methods: invoking, defining, parameters and arg-lists
- classes: defining, initialize, attributes
- method taxidermy: instance methods, class methods, singleton, self
- Ruby’s object oriented model
- object scoping
    - constants
    - globals
    - instance variables
    - class variables
- duck typing
- errors and exceptions
    - begin/rescue
    - raise
    - throw/catch
- access control
    - public, protected and private methods
    - aliases

(Meta)Mastery — Week 4

- Ruby review (kick it up another notch)
- advanced Ruby techniques
- intro to metaprogramming, reflection and introspection
- how Ruby finds and invokes methods
    - method search process
    - the missing method
    - hooks
- open classes and modules
- intro to Rake & Rails
- Ruby Gems
- metaprogramming use cases

PARSEC Ruby Bootcamp – Newbie – Week 1

	Monday	***Tuesday***	***Wednesday***	***Thursday***	***Friday***
9:00am ***Morning*** 12:00am	1. Introductions: instructor and students 2. Week/class overview 3. Setting the stage: ● Why are you here? ● Experience & expectations? ● What is Ruby? ● What's Ruby used for? 4. The Ruby programming language... from 5,000 feet ● Just enough Ruby to start... ● Reading Ruby code...	1. Teams – we all help each other... no one left behind! 2. Intro to Pair-Programming – ● All Labs are pair-pro... ● Team projects are bigger! **3. Lab – featuring:** **● Getting comfortable with the Sublime Text editor** 4. Ruby … getting started: ● puts (output) ● gets (input) ● Strings, interpolation ● Everything returns an object ● What's true? ...what's false?	1. Questions – no one gets left behind! 2. Gentle intro to OOPS... ● What is an Object ● Methods and Messages ● Data – *everything* is an Object! ● Why is this important? 3. Calling "subroutines" ● Method invocation ● What is a message? ● Arguments, parameters, and defaults ● return values	1. Questions – leave no teammate behind! 2. Intro to Git – ● Initializing your repository ● Checking in source code ● Repository status 3. The Ruby Library – ● Intro to the Library Routines ● Survey: the good ones... 4. Intro to Team Project **#1** ● Team organization – teams selected and named ● Discussion of project ● Clarifications ● Expectations ● Project setups	1. Questions – leave no teammate behind! 2. Team status reports (brief!) – ● Class feedback & help to each team... ● How is Ruby helping you to rapid-prototype and develop? 3. Team coding session...
	[Lunch Break]	***[Lunch Break]***	***[Lunch Break]***	***[Lunch Break]***	***[Lunch Break]***
1:00pm ***Afternoon*** 5:00pm	5. Team-building... What is collaboration? How do software development teams work? 6. Laptop setup – Linux, login, command line (bash), Sublime Text (editing), creating project directories 7. Resources: ● Laptops ● Linux & Command Line ● RVM & Ruby ● Sublime Text (editor)	5. Ruby ● Statements, expressions ● Flow control ● if/then/else and case ● loops ● iterators ● intro to Blocks 6. Ruby program structure – the big picture... 7. The name of the thing: naming Ruby programs, files, and more... **8. Lab – featuring:** **● REPL – irb** **● Scales and études**	4. Data in more depth... ● Data types? ● Duck typing! ● Ruby's built-in Data Objects: Numeric, String, Array, Hash ● Enumeration vs Iteration 5. Team development – ● Presenting to team ● Reading/understanding someone else's code... ● Code review etiquette ● How to work in and contribute to a coding team... **6. Lab – featuring:** **● Edit/test/debug cycle** **● Scales and études**	5. Team breakouts – ● Discuss approach and methodology ● Select roles: captain, scribe, archivist & spokesperson ● Discuss program design, how to get started... 6. Team coding session... 7. Session wrap-up – what's going well, what's in the way?	4. Team coding session... Wrap-up! (~90 minutes) 5. Team presentations of working code/programs to class... 6. Evaluations – ● Best design ● Best code/implementation ● Best documentation ● Best presentation ● Best teamwork

PARSEC Ruby Bootcamp – Foundational – Week 2

	Monday	*Tuesday*	*Wednesday*	*Thursday*	*Friday*
9:00am ***Morning*** 12:00am	1. Strings are powerful – ● Interpolation “#{s}” ● Searching and substitutions ● Other String lib-methods 2. Intro to RegExs (regular expressions) – ● What good is a regex? ● Pattern matching ● Regex compared to other string operations	1. Building a command line interface – ● by-hand, the hard way... ● with the OptionParser gem... ● making the Ruby program respond to com-line options 2. Configuration files – ● with the YAML gem... ● interaction with com-line options...	1. Questions – leave no teammate behind! 2. Intro to Team Project **#2** ● Team organization – new teams selected and named ● Discussion of project ● Clarifications ● Expectations ● Project setups	1. Questions – leave no teammate behind! 2. Team coding session... 3. Session wrap-up – what’s going well, what’s in the way?	1. Questions – leave no teammate behind! 2. Team status reports (brief!) – ● Class feedback & help to each team... ● How is Ruby helping you to rapid-prototype and develop? 3. Team coding session...
	[Lunch Break]	*[Lunch Break]*	*[Lunch Break]*	*[Lunch Break]*	*[Lunch Break]*
1:00pm ***Afternoon*** 5:00pm	3. More fun with Ruby Library – ● String class methods ● Numeric class methods ● Hash & Array methods ● File class methods ● Kernel & Process methods ● Finding the hard-to-find ones 4. Intro to Ruby Gems – ● What are “Gems”? ● Where do they come from? ● How are they organized? ● How to find “the good ones”? **5. Lab – featuring:** **● Scales and études**	3. Intro to Debugging – ● The print/puts method ● Responding to the com-line: ...if options[:verbose] and ...if options[:debug] ● Ruby debugger ● pry **4. Lab – featuring:** **● Scales and études**	3. Team breakouts – ● Discuss approach and methodology ● Select roles: captain, scribe, archivist & spokesperson ● Discuss program design, how to get started... 4. Team coding session... 5. Session wrap-up – what’s going well, what’s in the way?	***Guest Speaker*** ***Special Presentation:*** ***Working on a Ruby Project Team; your first Ruby job***	4. Team coding session... Wrap-up! (~90 minutes) 5. Team presentations of working code/programs to class... 6. Evaluations – ● Best design ● Best code/implementation ● Best documentation ● Best presentation ● Best teamwork

PARSEC Ruby Bootcamp – Essentials – Week 3

	Monday	***Tuesday***	***Wednesday***	***Thursday***	***Friday***
9:00am ***Morning*** 12:00am	1. Introductions: instructor and students 2. Week/class overview 3. Ruby Review – the basics are comfortable... ● Idioms 4. Methods ● Invoking a method ● Defining a method	1. Method taxonomy – ● Instance methods ● Class methods ● The Singleton method ● What is self 2. The Ruby OO-model ● Everything is an Object ● What's an Object? ● What's an Attribute? ● Modules and Classes ● Modules Mixin ● Class Inheritance ● Open Classes ● The Monkeypatch	1. Intro to Team Project **#3** ● Team organization – new teams selected and named ● Discuss approach and methodology ● Select roles: captain, scribe, archivist & spokesperson ● Discuss program design, how to get started... 2. Team coding session...	1. Questions – leave no teammate behind! 2. Team coding session... 3. Session wrap-up – what's going well, what's in the way?	1. Questions – leave no teammate behind! 2. Team status reports (brief!) – ● Class feedback & help to each team... ● How is Ruby helping you to rapid-prototype and develop? 3. Team coding session...
	[Lunch Break]	***[Lunch Break]***	***[Lunch Break]***	***[Lunch Break]***	***[Lunch Break]***
1:00pm ***Afternoon*** 5:00pm	5. Methods (continued) ● Parameters – defining – default values – aggregates (splat) – keyword (hash) – variable length arg-lists – block 6. Classes – ● foo = Foo.new(…) ● The initialize method ● Class attributes and attr_* macros	3. Object Scoping ● Constants & Globals ● Instance Variables ● Class Variables 4. Duck typing 5. Errors and Exceptions ● begin/rescue/... ● raise ● throw/catch 6. Access control ● Public methods ● Protected methods ● Private methods ● Aliases **7. Lab – featuring:** **● Scales and études**	3. Team coding session... 4. Session wrap-up – what's going well, what's in the way?	***Guest Speaker*** ***Special Presentation:*** ***What Does It Take to be a Great Software Developer***	4. Team coding session... Wrap-up! (~90 minutes) 5. Team presentations of working code/programs to class... 6. Evaluations – ● Best design ● Best code/implementation ● Best documentation ● Best presentation ● Best teamwork

PARSEC Ruby Bootcamp – (Meta)Mastery – Week 4

	Monday	***Tuesday***	***Wednesday***	***Thursday***	***Friday***
9:00am ***Morning*** 12:00am	1. Introductions: instructor and students 2. Week/class overview 3. Ruby Review – The basics are mastered... 4. Advanced Ruby ● Lots of code examples	1. Rake – Ruby's answer to make... on steroids 2. Code walkthroughs: ● Sample Gems ● Rake idioms ● Building and submitting a Ruby Gem	1. Intro to Team Project **#4** ● Team organization – new teams selected and named ● Discuss approach and methodology ● Select roles: captain, scribe, archivist & spokesperson ● Discuss program design, how to get started... 2. Team coding session...	1. Questions – leave no teammate behind! 2. Team coding session... 3. Session wrap-up – what's going well, what's in the way?	1. Questions – leave no teammate behind! 2. Team status reports (brief!) – ● Class feedback & help to each team... ● How is Ruby helping you to rapid-prototype and develop? 3. Team coding session...
	[Lunch Break]	***[Lunch Break]***	***[Lunch Break]***	***[Lunch Break]***	***[Lunch Break]***
1:00pm ***Afternoon*** 5:00pm	5. Intro to – ● Metaprogramming ● Reflection & Introspection 6. How Ruby finds & invokes methods ● The method search process ● The "missing method" 7. Open Classes & Modules: the Monkeypatch and the Mix-In revisited	3. Metaprogramming use cases 4. Code walkthroughs: ● Rails idioms	3. Team coding session... 4. Session wrap-up – what's going well, what's in the way?	***Guest Speaker*** ***Special Presentation:*** ***Participating in an Open Source Project***	4. Team coding session... Wrap-up! (~90 minutes) 5. Team presentations of working code/programs to class... 6. Evaluations – ● Best design ● Best code/implementation ● Best documentation ● Best presentation ● Best teamwork