BUS36110

Application Development

> Knowing how to hack also means that when you have ideas, you'll be able to implement them… It's a big advantage, when you're considering an idea like putting a college facebook online, if instead of merely thinking "That's an interesting idea," you can think instead "That's an interesting idea. I'll try building an initial version tonight."
>
> — Paul Graham

Why

The new reality is that every company is a software company. Even in traditionally brick-and-mortar industries, software is performing more and more of the work. And, of course, many companies (especially "lean startups") are purely software-based.

Lacking an understanding of how software works while working in a software world puts you at a disadvantage. In this course, you will build a functional prototype of a software application. To do this, you will learn the Ruby on Rails web application framework, but our higher-level goals are to:

1. Understand the general, platform-independent patterns of how applications work.
2. Communicate more effectively and credibly with developers on our teams.
3. Develop a builder's eye for problems that can be solved with technology.
4. Prioritize features more intelligently by developing a better feel for their costs.
5. Be able to make and test small changes to an application yourself.
6. Become familiar with the agile development technique.
7. Be empowered with a new, powerful, and just plain fun creative outlet

Who, Where, & When

Instructor:
Raghu Betina (Raghu.Betina@chicagobooth.edu)
Lecture:
Wednesdays 6:00-9:00pm, Booth 455 NBC Towers
Office Hours:
TBD. Considering Google Hangouts

Required Materials

Text:
Chris Pine's *Learn to Program, Second Edition*.
Software:
You will receive a detailed guide to setting up a development environment on your Mac or Windows laptop. All required software is free. In addition, some optional tools that will make your life easier have a free trial.

Course Format

This course is entirely project-driven. We will build a series of applications together in class. We will start with a simple idea and learn just what we need about various technologies (HTML, CSS, Ruby, Rails, JavaScript) to make that idea real.

We will then build a slightly more complicated idea, and deepen our knowledge of each topic. And then an even more complicated idea. In this way, we will spiral across topics picking up essential, applicable knowledge; what we're not going to do are deep theoretical dives into each topic, one after the other, without ever getting to build real stuff.

In addition, you will build your own app idea. During the first few weeks, you will meet with an instructor to discuss your ideas and select one that matches the scope of the course. Each week, part of your homework will be to plug the concepts you learned in class into your app. This app will be your final project.

Lectures will generally consist of the instructor demonstrating a concept and then giving the class a challenge to practice it and think up questions. You are strongly encouraged to work in pairs during these challenges.

Expectations & Time Commitment

This course is designed for a beginner who has never programmed before. It is our goal to make programming accessible to as many people as possible, and dispel the mystique that surrounds it. It's really possible, with passion and persistence, for a beginner to become productive very quickly with today's tools.

Note: it's *possible*, but not always *easy*. If you have never seen a line of code before, you are going to be learning a lot of new things all at once, and so should plan to spend a significant amount of time. If you already have some programming experience, you may have to spend less time initially while we learn basics; but once we get into Rails, it will likely be all new to you too.

In short, it is not recommended to take this course if you are a completely new to programming *and* you have a lot of other commitments this quarter.

What you can expect from us in return is a lot of support. We know it's a lot, and we're here for you -- one-on-one coaching appointments, office hours, etc. We all will help each other on the class discussion board. There is no such thing as a dumb question; we're all beginners here, learning together.

Attendance (10% of your grade)

Lecture attendance is mandatory, and you are expected to be on time. To account for the vagaries of life, you will be allowed to miss one class with no penalty. After that, 1% will be deducted from your final grade for each unexcused absence. In addition, you will not gain access to the lecture recording of any class you do not attend.

An excused absence is a documented medical emergency, religious holiday, etc. Get in touch with the instructor immediately if you foresee any such absences.

Attendance of coaching appointments and office hours will be noted and may make the instructor more sympathetic when it comes to borderline grades.

Homework (40% of your grade, 5% each)

Homeworks will be posted each week by noon on Thursday. Homeworks will generally consist of re-doing what we did together in lecture, applying that concept to a few other contexts, and then applying it to your own app idea.

Homeworks will be due on Wednesday at 5pm; no exceptions, because we will often discuss homework solutions in class. To account for the vagaries of life, you can miss one homework submission without penalty.

If you want any partial credit on homeworks, you must schedule a one-on-one appointment with a Faculty Coach to go through your solutions and discuss anything that's fuzzy. You can book up to one 30 minute appointment per week. The appointment can be remote if necessary, via Google Hangout. We strongly recommend taking advantage of this unique resource; we are lucky to have professional developers acting as Faculty Coaches.

You are allowed and encouraged to work in study groups on the homeworks, but you must type up your final solutions yourself. In other words, no copy-pasting.

Midterm (25% of your grade)

The midterm exam will occupy one-half of Week 7 class. It will be a practical exam; you will download broken applications and fix them.

Final Project (25% of your grade)

The final project will be to build one of your own ideas and to write a brief essay about your app and what you learned from the process of building it. Note that if you complete all your homeworks on time, your final project should be well underway by Week 9, and after that you just need to add polish.

Tentative Schedule (May change to accommodate guest presenters & student needs)

Week 1	**Getting Started**
	Course introduction
	Text editor basics & keyboard shortcuts
	GitHub basics
	HTML & CSS
	Goal/HW: Push a webpage to GitHub, read "How to Get Startup Ideas"
	Defining Your MVP
	Bootstrap & tools
	Low-fidelity sketches

Week 2	Extracting user stories
	Prioritizing and eliminating non-essentials
	Domain Modeling
	Goal/HW: Settle on an MVP, mock up idea in HTML, diagram domain model
	Ruby Foundations
	Data types & instructions (begin reading Pine book)
Week 3	Managing lists of things
	Defining our own instructions
	APIs
	Goal/HW: Finish Pine book
Week 4	**Ruby Objects**
	Defining our own data types
	Creating individual instances of objects
	Adding attributes
	Class inheritance
	Goal/HW: Read data from an external service like Facebook
	Rails Fundamentals
	Setting up a web server
	Listening for requests
	Responding dynamically
	Goal/HW: Get a web server up and running, add your static HTML mockups
Week 5	**Console CRUD**
	Database fundamentals
	Creating, reading, updating, and deleting
	Receiving inputs from the user
	Goal/HW: Add one database table
	Associations
	One-to-many
	Goal/HW: Add more models and relate them to one another
Week 6	**Domain Modeling**
	Many-to-many
	Goal/HW: Translate user stories into queries
	Web CRUD
	Wiring up HTML forms
	The Golden Seven
	Goal/HW: Complete all database-backed web resources
	Interface
	MIDTERM

Week 7	Displaying associated objects
	Goal/HW: Satisfy user story interface requirements
Week 8	**Authentication and Authorization**
	Cookies
	Encryption
	Filters
	Goal/HW: Add sign-in/out and personalized pages
Week 9	**Extra Credit**
	Validations, scaffold roundup
	Gems, pagination
	Deployment
	Callbacks
	Goal/HW: Implement one of the above
Week 10	**Fun Stuff**
	Sending email and SMS, payments, JavaScript, file uploads, testing
	Resources for further learning
	No homework
Week 11	**Submit Final Project**
	Celebrate!