

# Course Syllabus

## Locations and Instructors

|  |  |
| --- | --- |
| CHARLOTTE CAMPUS 8430 University Executive Park Dr #650 Charlotte, NC 28262 704.913.3077 | TRIAD CAMPUS 1231 Shields Rd Suite 5 Kernersville, NC 27284 336.231.8632 |
| Andrew Jensen *(Director of Education)* <http://andrewjensen.coderfoundry.com> [ajensen@coderfoundry.com](mailto:ajensen@coderfoundry.com) | **Thomas Parrish** <http://thomasparrish.coderfoundry.com> [tparrish@coderfoundry.com](mailto:tparrish@coderfoundry.com) |
| Antonio Raynor [http://antonioraynor.coderfoundry.com](http://Andrewjensen.coderfoundry.com) [araynor@coderfoundry.com](mailto:araynor@coderfoundry.com) | **Ria Manglani** <http://riamanglani.coderfoundry.com> [rmanglani@coderfoundry.com](mailto:rmanglani@coderfoundry.com) |

## Prerequisites

1. Four-year degree in Computer Science, Information Systems, or related field

**OR**

Completion of the following:

* Ten-question Logical Reasoning Assessment
* Ten-question Abstract Reasoning Assessment
* Eight-question Numerical Reasoning Assessment
* Satisfactory completion of a series of JavaScript coding exercises
* Applicants lacking experience or training in fundamental computer science and programming are required to complete the following free tutorials:

|  |
| --- |
| Tutorial Resources |
| HTML and CSS Fundamentals (MSVA) <http://www.microsoftvirtualacademy.com/training-courses/html5-css3-fundamentals-development-for-absolute-beginners> |
| HTML and CSS Fundamentals (MSVA) <http://www.microsoftvirtualacademy.com/training-courses/html5-css3-fundamentals-development-for-absolute-beginners> |
| JavaScript Fundamentals for Absolute Beginners (MSVA) <http://www.microsoftvirtualacademy.com/training-courses/javascript-fundamentals-for-absolute-beginners> |
| C# Fundamentals for Absolute Beginners (MSVA) <http://www.microsoftvirtualacademy.com/training-courses/c-fundamentals-for-absolute-beginners>  C# Fundamentals (Pluralsight) <http://www.pluralsight.com/courses/csharp-fundamentals-csharp5>  Object Oriented Fundamentals in C# (Pluralsight) <http://www.pluralsight.com/courses/object-oriented-programming-fundamentals-csharp> |
| Database Fundamentals (MSVA) <http://www.microsoftvirtualacademy.com/training-courses/database-fundamentals> |
| Software Development Fundamentals (MSVA) <http://www.microsoftvirtualacademy.com/training-courses/software-development-fundamentals> |

*\*MSVA = Microsoft Virtual Academy*

1. All Coder Foundry applicants are required to complete the free Pluralsight tutorial, *Becoming a .NET Developer* (<http://www.pluralsight.com/courses/becoming-dotnet-developer>), prior to the session start date.

### Required at start of class:

1. Personal laptop meeting the following requirements:
   * 2.0 GHz or faster processor
   * 6 GB RAM or better (8+ GB is recommended)
   * HDMI input
   * Windows 7 or newer operating system (a MacBook running Windows in a virtual environment such as BootCamp is acceptable, provided the above requirements are met)

Machines meeting these requirements may be purchased at Wal-Mart, Best-Buy, Amazon.com, and other retailers for as little as $340.

1. Software installations (all free versions):
   * Visual Studio 2013 Community with Update 4 or 5   
     <http://www.visualstudio.com/downloads/download-visual-studio-vs>
   * SQL Server Express 2014 with Tools  
     <http://www.microsoft.com/en-us/download/details.aspx?id=42299>
   * Google Chrome (we prefer this browser for its on-board debugging tools)
   * Git desktop client (we prefer either GitHub for Windows or GitExtensions)  
     <https://github.com/>   
     <http://sourceforge.net/projects/gitextensions>
2. Personal requirements:
   * Professional e-mail account   
     [thorwisheshewasme@gmail.com](mailto:thorwisheshewasme@gmail.com), [teddybearsrule@yahoo.com](mailto:teddybearsrule@yahoo.com), and [onehotnerd@hotmail.com](mailto:onehotnerd@hotmail.com) are NOT professional email accounts
   * Professional resume (at least a draft)
   * Github account  
     <https://github.com/>
   * Microsoft online account  
     <https://signup.live.com/>

## Course Description

This is a fast-paced, interactive learning experience that employs a learn-by-doing theory of education. Lectures are typically short in duration – 30 to 60 minutes – during which essential skills related to the next development project are taught. Lecture sessions are followed by extensive and intense coding sessions focused on solving real-world problems. Upon completion you will have completed in excess of 600 hours of coding during which you will have built multiple real-world projects, all of which will be made available for viewing by prospective employers on your personal Website.

You will be treated as an “employee” of Coder Foundry during your time in the course, with project specifications, weekly project deliverables, deadlines, and accountability interviews.

Mondays are reserved for Accountability Interviews, during which you will report on the work you have completed during the previous week, your status on the current deliverable and project, and any pitfalls or difficulties you may have encountered. You will also demo your software in its current deliverable state, and may be asked to discuss various aspects of the code you have written to that point. In addition, these interviews serve as an opportunity for us to coach you on valuable interviewing skills in an effort to better prepare you for the job interviews in which you will participate upon completion of the course.

### Course Pathways

Not all students enter Coder Foundry with the same level of development skill, or with the same preparation. To better accommodate all of our students, we have incorporated two distinct curriculum pathways within our Master Class.

Upon completion of the course prep work and personal interview, the Coder Foundry staff will assess each student’s overall progress and ability, and assign that student to the pathway that bests suits his or her individual situation.

Once assigned to a specific pathway, a student need not remain in that pathway for the duration of the course. Each Monday interview *during the first three weeks* of the course is an opportunity for Coder Foundry staff to reassess each student’s progress, and thus determine whether the student’s current pathway is the best fit to ensure that student’s continued success.

Every student who attends Coder Foundry should understand that our primary objective is to help him or her find and maintain, through the teaching and application of marketable skills, gainful employment as a software developer. The job market is not like an academic institution, where an inadequate C-minus grade is considered passing. No one gets or keeps a job with a C-minus performance as a developer. It is far better to be an outstanding developer in one area (like MVC) than merely adequate in multiple areas.

#### API Pathway

This is our standard curriculum plan. It assumes each student possesses a certain level of development experience sufficient for success in the course without supplementary instruction. It includes training in HTML5, Bootstrap CSS, JavaScript, C#, ASP.NET MVC, ASP.NET Web API, and AngularJS.

##### Course Schedule

|  |  |  |
| --- | --- | --- |
| Week | Topics | Project(s) |
| 1 | Git  Visual Studio  Bootstrap  JavaScript/jQuery review  MVC (introduction)  Microsoft Azure | Bootstrap Exercises  JavaScript Exercises  Project: Personal Website |
| 2 | Object-oriented programming in C# MVC (authentication and authorization, view models vs.  data models, Code First database development, LINQ) | MVC Project #1 |
| 3 | MVC (partial views, paging, search and filtering, More LINQ) |
| 4 | MVC (Scaffolding, More LINQ, databases) | MVC Project #2 |
| 5 | Project Development |
| 6 | Project Development |
| 7 | SQL and Stored Procedures  Web API (controllers, routing, token-based authentication) | Web API / AngularJS Project #1 |
| 8 | AngularJS |
| 9 | AngularJS | Web API / AngularJS Project #2 |
| 10 | AngularJS  Project Development |
| 11 | Project Development |
| 12 | Project Development |

#### MVC Pathway

This pathway is designed for those students who lack formal computer science training or significant coding experience. This course returns to the fundamentals of computer programming theory and constructs. Student learning is focused on HTML5, Bootstrap CSS, JavaScript, C#, and ASP.NET MVC. This pathway dispenses with the Web API and AngularJS portions of the course.

##### Course Schedule

|  |  |  |
| --- | --- | --- |
| Week | Topics | Project(s) |
| 1 | Git  Visual Studio  Bootstrap  JavaScript/jQuery review  MVC (introduction)  Microsoft Azure | Bootstrap Exercises  JavaScript Exercises  Project: Personal Website |
| 2 | Programming Fundamentals in C#  Object-oriented programming in C# | MVC Project #1 |
| 3 | MVC (authentication and authorization, view models vs.  data models, Code-First database development, LINQ) |
| 4 | MVC (user management, role assignment) | MVC Project #2 |
| 5 | MVC (scaffolding, more LINQ, databases) |
| 6 | MVC (paging, search and filtering, more LINQ, databases) |
| 7 | MVC Review (previous topics) | MVC Project #3 (optional) |
| 8 | Project Development |  |
| 9 | MVC (partial views) | MVC Project #4 |
| 10 | Project Development |
| 11 | Project Development |
| 12 | Project Development |

##### MVC Pathway Supplementary Material

Items in this section may be assigned to individual students during the first three weeks of the course if the instructor feels the student needs additional instruction in web programming fundamentals below the level at which the curriculum is designed to operate. Current recommendations are the following free online courses provided by Codecademy™.

|  |  |
| --- | --- |
| Course Title | URL |
| HTML & CSS | <https://www.codecademy.com/en/tracks/web> |
| JavaScript | <https://www.codecademy.com/en/tracks/javascript> |
| jQuery | <https://www.codecademy.com/en/tracks/jquery> |

Students may also be asked to complete or repeat the course prep work, found at the beginning of this syllabus, if their preparation is deemed insufficient.

**Note:** We reserve the right to alter course schedules to best meet the needs of the class body, as determined by the instruction team.

## Grading

We follow a grading pattern equivalent to that of an employer, as opposed to grading systems used in academic settings. All student projects will be assigned a Pass/Fail grade based upon strict adherence to project specifications, deliverables, and deadlines. Simply put, students who complete each project *on time and as required by the project specification* receive a passing grade for that project.

The course curriculum consists of five (5) total projects of varying degrees of difficulty.

1. Personal Web Site and Portfolio
2. MVC Small Project
3. MVC Large Project
4. Web API / AngularJS Small Project
5. Web API / AngularJS Large Project

Projects 4 and 5 will be additional MVC projects for students completing the MVC Pathway.

To receive a passing grade for the course, students must receive passing grades for four (4) of the five (5) projects, including projects 1, 3, and 5.

We actively work to help students *who receive passing grades* obtain employment by seeking out suitable job opportunities on their behalf, setting up job interviews, and providing interview coaching.

*We cannot adequately market non-passing students to potential employers*. Students must have a *suitable body of work* that we can show potential employers before we can schedule interviews. A suitable body of work is the result of a passing grade for the course.

## Textbooks

We pride ourselves in our ability to sit at the leading edge of software development technologies, and do not make use of printed (read: quickly outdated) textbook materials. We do make use of a variety of online reference sources, which are updated from one course to the next as required.

We also provide the following quick reference guides for students, as their content is not as quick to change as typical language resources.

* C# 5.0 Pocket Reference   
  Joseph Albahari and Ben Albahari  
  ISBN-13: 978-1449320171
* LINQ Pocket Reference  
  Joseph Albahari and Ben Albahari  
  ISBN-13: 978-0596519247

## Class Policies

All Coder Foundry students are expected to conduct themselves in a professional and respectful manner at all times. As a professional technology organization, we strongly adhere to the [ACM/IEEE Code of Ethics and Professional Practice](http://www.acm.org/about/se-code#short), and we expect the same conduct of our students. Students who knowingly and consistently act in a manner that violates or opposes this code of conduct will be dismissed from the course. In addition, the following policies are specific to Coder Foundry classroom operations.

### Attendance

Coder Foundry is not designed to operate as an academic institution, but rather as a workplace simulation. Attendance is mandatory, just as it is on a job. The average company provides employees with two (2) weeks (ten (10) business days) or personal paid time off. We permit students four (4) total absences during the 12-week course. That’s equivalent to 16 days of paid leave in the average job, far more than an employee would reasonably expect. Coder Foundry does not provide “make-up” opportunities for absences.

Students who accrue more than four (4) absences for any reason may be dropped from the course. Absences, regardless of reason, have no effect on project deadlines. A project that is incomplete or late is a failed project.

Emergency situations such as serious illness or hospitalization, family deaths, and other extenuating circumstances are considered exceptions, and are dealt with on a case-by-case basis. Such occurrences require the presentation of documentation verifying the exceptional nature of the circumstance.

### Tardiness

As on the job, tardiness is not permitted. It disrupts the teacher as well as the other students and inhibits the learning environment. A student who is more than fifteen (15) minutes late for the day’s session will be considered absent for the entire day. Such an occurrence will count toward the student’s allotted four (4) absences for the course.

### Classroom Behavior

Disruptive behavior is unacceptable in any classroom and in any workplace. Students will not engage in non-class-related behavior, such as other employment activities, telephone conversations, video conferencing, online shopping, social networking, online videos or movies, video gaming or gambling, grooming, napping, or other behavior that is inappropriate to a work environment while in class. Students may be asked to leave for the day and considered absent should such behavior occur. Persistent behavior of this type will result in dismissal from the course. Dismissed students are not entitled to a refund of any portion of their tuition fees, as a dismissed student is not classified as a *withdrawn* student, and is therefore not subject to the terms set forth for student refunds (see Financial Information, page 17).

*Respectful* heckling of instructors is permitted in moderation.

### Personal Integrity

Academic and professional integrity are of the utmost importance. Your work must be your own. While we encourage our students to assist and learn from one another, just as they would in a work environment, ultimately each student is responsible for his or her own work. Submitting a project that is someone else’s work is absolutely unacceptable and will result in immediate dismissal from the course.

Students should understand that the coding profession is unlike that of a writer, in that it is common for coders to borrow from one another’s solutions to specific coding problems. For example, if a student desires a solution for client-side pagination with AngularJS and is unsure how to derive such a solution independently, we encourage that student to research possible solutions. Doing so may lead the student to a coded solution on any one of a variety of resource websites. Borrowing others’ coding solutions to specific problems is not plagiarism. Borrowing another person’s project, or significant portion thereof, is. We encourage our students to use specific solutions that they discover through their own research, but also to endeavor to understand the solutions they choose to borrow. If a student cannot articulately explain his or her own code in a progress interview, that student will not pass the project.

### Special Accommodations

If you have specific needs as the result of any disability, inform a member of the Coder Foundry staff before beginning the course so that necessary accommodations can be made.