A New Set of Hands-on Courses in Computer Science

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Bernard Dinh

Fall, 2022

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Briana Morrison, Department of Computer Science

A New Set of Hands-on Courses in Computer Science

CS 4991 Capstone Report, 2022

Bernard Dinh
Computer Science
University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
bd4jqg@virginia.edu

ABSTRACT

University of Virginia students would benefit from being taught more marketable skills to prepare them for the real world and increase their prospects of landing internships and jobs. I propose a new set of courses that will allow students to get hands-on experience working with common frameworks and techniques in the computer science field. The courses are structured in a similar way: composed of learning the core concepts of the framework and then applying these concepts in a semester long project. One core theme for these courses is being well structured, providing abundant resources for students to become acquainted with a specific technology. The course objective is to teach the fundamentals for a framework so that students can utilize this new technology in their future career. Additional adjustments should be made to these courses to facilitate independent and honest learning.

1. INTRODUCTION

The current Computer Science curriculum offers a wide variety of courses focused on different topics in the computer science field. The required core CS classes teach foundational knowledge such as common data structures and algorithms. Then, there are the elective courses which provide a broad overview of a particular CS discipline such as machine learning or computer networks. Courses can be project-based and revolve around building a semester-long project using the course concepts.

The content in many CS courses is more on the theoretical side and exercises students' knowledge of course material. However, I propose more emphasis on providing students with hands-on experience working with practical tools that they would likely encounter in the CS workplace. While theoretical coursework is effective at exercising critical thinking and reasoning skills, being able to apply this knowledge in a more direct way is important as well.

A problem is that during the semester, if students are focused on coursework and other endeavors outside of the classroom, they will have less time for training their skillset. To mitigate this, I propose a way for students to learn new tools/skills which they can then include on their resume, while concurrently gaining credits towards their degree. This relieves students of having to prioritize between personal development and academic work. It also expands students' toolbelt and better prepares them when applying for jobs.

2. RELATED WORK

The CS 3240: Advanced Software Development Techniques course offered at UVA inspired the design for this new set of courses (McBurney and Sherriff, 2022). This course has students form teams and go through the software development life cycle to create a functioning web application with Django. By the end of the semester, students become familiar with developing in Django and have a project to show for it. Lectures are centered on how software engineering works in real world. The proposed set of courses are similar in that they focus on a course project using a particular framework, but the lectures would focus solely on how to use that framework rather than complementary material.

The structure and content of coding tutorials found online (theodinproject.com, 2022; freecodecamp.org, 2022) roughly show how these courses would teach the fundamentals for a framework. Students would work through part of a tutorial every week and complete practice questions/tasks to ensure that they are keeping up. However, courses would include more resources on the subject and provide students with a direct way to ask questions.

3. PROPOSED COURSE DESIGN

This following section describes the general layout for the proposed courses in the form of a course syllabus. Frameworks/tools that are commonly used in the industry will be the main subject of these courses. These can be anything from web frameworks such as Ruby on Rails or front-end libraries such as React.

3.1 Prerequisites

Students should have taken CS 2150 and have a solid understanding of object-oriented programming. They should have a basic understanding of the core programming languages that the framework of the course relies on. Students should also be capable of independent learning.

3.2 Course Objectives

1. Develop an understanding of how to implement a project with the framework/technology.
2. Comprehend the fundamentals of using that framework involving syntax, project structure, use cases.
3. Be able to evaluate the benefits and limitations of the framework and understand the situations that it is best suited for.
4. Produce a functioning software project to show to future employers.

3.3 Coursework

Students will be required to complete multiple tutorial quick checks, quizzes, and a course project which constitutes most of the course grade.

3.3.1 Tutorial Quick checks (30% of grade)

Each week, students will work through a module of the framework tutorial hosted on the course website and demonstrate understanding of the material through weekly assignments. These assignments involve answering concept questions and doing small coding exercises relating to the content covered in the tutorial that week.

3.3.2 Quizzes (20%)

There will be two open-note quizzes to assess students' level of understanding of the content so far. These will involve multiple choice and short answer questions relating to language syntax, code snippets and their output/structure, and general concepts.

3.3.3 Project (50%)

The major deliverable in this course is the project that utilizes the framework. The two components that makes up project grading are the sprints worth 30% and the project demo at the end of the semester worth 20%.

3.4 Project

For the given framework, students will have 3 project options to choose from, each with their own set of requirements to incorporate. The options will be formulated to implement all the material learned in the tutorial while also allowing students flexibility in customizing the product. Examples of projects include a blog site, creating an Instagram clone, online marketplace, etc.

Students will be graded on the quality of their final product and whether they made acceptable progress for each sprint. Aspects that will affect the quality include the appearance of the UI, existence of major bugs, practicality, and overall polish.

3.4.1 Sprint checks (30%)

Students will be required to meet with TAs every two weeks (totaling 3 sprints) to demonstrate progress on the project and have the chance to ask questions. There will be a set of features/requirements that need to be implemented before the end of each sprint. Roughly the same amount of work will be assigned for each sprint, and students will be graded on completion of the features i.e. not accounting for aesthetics.

3.4.2 Demo (20%)

Students have the option of either creating a 3-5 minute video or producing a writeup, demonstrating all the features of the final product. In either option, they will discuss limitations and possible improvements of the final product. Writeups, videos, and URL links to the product (if applicable) will be posted for other students to view, comment, and vote on. The top three projects will receive five extra points on their project grade as incentive for students to build a high-quality product.

3.5 Logistics

Each course will have a web tutorial to complete, after which students will work on the course project. Each course will operate on a Tuesday/Thursday 2-3:15 schedule and the class size will be 60.

3.5.1 Schedule

The courses will follow a 15-week schedule. The first half of the semester from Weeks 2-7 will be students working through the tutorial and completing exercises to gain familiarity with the framework. Quizzes will be administered in Weeks 4 and 7. The rest of the semester will be for completing the project with sprint checks at weeks 9, 11, and 13. The end of Week 14 will be the deadline for submitting project demos. During Week 15, students can view the gallery of projects and vote on their favorites.

3.5.2 Lectures

Lectures will be instructors reteaching the tutorial lessons for that week, providing additional clarifications and insights. Lectures will be recorded, and attendance will not be mandatory.

3.5.3 Collab

Collab will contain links to recorded lectures, grades, and overall course information and be the site for submitting assignments.

3.5.4 Course Website

The course website will host the tutorial and exercises that students will have to work through. It will contain helpful resources that pertain to the framework. At the end of semester, the project gallery of student projects will also be located here.

3.5.5 Piazza and Discord

Piazza will be for answering general framework questions or logistics questions. Office hours will be held on discord voice channels which allows screen sharing.

3.6 Honor policy

Students are not allowed to share code with one another and can only speak generally about their project with other students. Students must cite code they found online.

4. ANTICIPATED RESULTS

In passing the course, students should come away with a decent understanding of a framework and be able to design and implement similar projects. They will have produced a functional project and can put this skill on their resume. In the future, they can reference the notes, resources, and tutorial in this course.

5. DISCUSSION AND REASONING

Students will have limited project options with a common goal of promoting independent learning.

5.1 Project Format

Students would be provided options for projects rather than letting them come up with the project ideas themselves. This will allow grading to be more streamlined and ensure that all the framework concepts are tested. Each option will have a set grading rubric and its own requirements, whereas student designed options will vary in difficulty and may not test the framework concepts in good detail. Limited project options will also make it easier for course staff to debug and assist students in office hours.

While software development is usually done in groups in the industry, the projects will be done independently in these courses. The reason is to give each student an equal exposure to the framework concepts. This is because it is often the case that group members may do a disproportionate amount of the workload and so some members will come out with an incomplete understanding of the concepts. Therefore, although the primary goal of these courses will be to prepare students for the industry, the focus will be getting students acquainted with the technical side of things rather than the social side (i.e., group dynamics).

5.2 Goals

These courses will promote independent learning by pooling together resources to facilitate this learning, including Piazza, office hours, tutorial exercises, and supplementary lecture content. This provides students with ample opportunities to ask clarifying questions and debug more easily compared to working outside the course where finding assistance can be more difficult because web resources are sparser.

6. CONCLUSION

One of the main purposes of attending university is for students to learn foundational skills that will prepare them to enter the workforce. To fulfill this purpose, the proposed set of courses will provide computer science students the resources necessary to learn a technical framework in a controlled environment. In this way, they obtain more experience by acquiring another tool which they might utilize in future endeavors.

7. FUTURE WORK

One challenge that needs to be overcome is designing unique project options that sufficiently test the tutorial concepts. Project options would also have to differentiate from popular project ideas that already exist online. This is to encourage more original solutions and implementations. Another area for improvement is offering more project options for students to pick a project that they will likely put more effort into. However, this would require more course staff as there would be a wider variety of implementation questions and bugs to handle in office hours and piazza.

REFERENCES

McBurney, W. and Sherriff, M. 2022. An Introduction to Software Engineering. Retrieved September 25, 2022 from https://www.cs3240.org/

theodinproject.com. 2022. JavaScript | The Odin Project. Retrieved September 25, 2022 from https://www.theodinproject.com/paths/full-stack-javascript/courses/javascript

freecodecamp.org. 2022. Front End Development Libraries. Retrieved September 25, 2022 from https://www.freecodecamp.org/learn/front-end-development-libraries/