

Paul Nickerson
COP3530



“Paul Nickerson” sounds like Pall (rhymes with tall) Nick-ur-sun. I just go by Paul.

All About Me

I am driven by a desire to make a positive impact on the world. What some may perceive as an unhealthy degree of selflessness I view as a responsibility to use my gifts - creativity, intelligence, and empathy - towards improving the lives of others. It is this trait that led me towards leaving the startup world to finish a degree in biochemistry; I wanted to apply my skills in software engineering to the medical field in order to help pioneer technologies that save lives and improve quality of life (it turns out doing so is not nearly as straightforward as I had naively anticipated, so I am currently reassessing my options. C'est la vie). It is also why I registered for this class in the hopes that the experience would improve my engineering skills, despite several warnings that doing well would require sacrificing precious time which could be spent enjoying my final college semester. For most of this year, I have been involved with research at the biomedical engineering department applying data mining and machine learning efforts towards problems with medical significance, generally related to developing smarter surgical and anaesthesiological methods.

As the eldest of two sons, I have a responsibility to be a role model for my younger brother, Joe. We are very close; in fact, we share a house with two roommates. Our family grew up in Gainesville, so attending UF was always somewhat of an implicit expectation. My childhood was relatively tumultuous. Our parents divorced when I was very young, and they struggled to resolve their disputes for many years, which frequently led to my brother and me getting caught in the middle of difficult "grownup problems". According to the Diagnostic and Statistical Manual of Mental Disorders, I have a pretty strong case of ADHD. As a result, my relationships with teachers and peers involved plenty of friction; however, I believe this "disorder" also endows me with an abundance of creativity for which I am thankful. The Myers-Briggs test reports my personality as being mildly extroverted and moderately strong intuitive, feeling, and perceptive (ENFP). As such, I tend to become very enthusiastic about a wide variety of different things. While I have a tendency towards chaos, my value system is acutely developed and motivates my passion.

These days, schoolwork and research take up most (all?) of my time. I try to regularly make it to the gym and lift weights, which is something I really enjoy; although, summer semester coursework required an inordinate amount of work and prevented me from exercising as often as I would have preferred. When I have spare time and money, I love traveling and hiking, preferably at the same time. The photograph included on the first page shows me posing alongside my brother in the mountains of North Carolina. Leisurely reading happens occasionally when I convince myself that I need to be more scholarly and cultured. My favorite show of all time is Doctor Who (followed closely by Arrested Development, Archer, and Sherlock in no particular order). Of the four Doctors I have seen since the show's resurrection in 2005, the tenth - played by David Tennant - is my favorite. I closely identify with the Doctor as the lonely outsider, the last of his species, lovingly observing time and space and making a positive impact on the lives of its inhabitants when he can while presenting a goofy and absent-minded persona.

Course Expectation

As Dave explained in his description of this course, Data Structures and Algorithms is somewhat of a learning “bottleneck,” containing so much information crucial to higher-level computer science topics that, as the only programming course I will have had the chance to take during my undergraduate years, it offers the best opportunity to receive formal computer science training before either returning to the workforce or considering a graduate degree. I hope to learn a wide range of concepts that I can apply to improve my skills as a software engineer as well as to provide a foundation upon which I may further my education.

Dave should expect from me a commitment to put forth my best effort as a student; I am taking this course not to satisfy some requirement (although I am hoping to earn the highest grade possible for the sake of my GPA), but rather to receive the best computer science education that I can in a single semester. I expect that Dave’s reputation as an incredibly competent teacher is well-deserved and that he can explain concepts and provide grade incentives in such a way that maximizes the learning outcome.

An excellent course is one that helps students to conceptually think about the material as one who works in the field would. Towards this end, it is essential to provide students with plenty of opportunities to practice the material by applying it to solve problems they might realistically encounter. Test scores should reflect the degree to which students *intuitively* understand the concepts taught. As a rule of thumb, a course’s excellence is inversely proportional to the amount of facts which must be memorized and regurgitated on exams, only to be later forgotten.

Programming Background

I have been writing code for about 15 years; although I make a distinction between the webpages I was writing during my early developer years and actually writing software, which I have been doing for about 10 years. After building a few Pokemon fan sites starting in 4th grade, I got involved in security research and penetration testing. During that time I was mainly identifying vulnerabilities in various server and client applications as well as web browsers, and software I was writing was meant to either facilitate the process of finding bugs or to show their exploitation via proofs of concept. During high school, my contributions to the field of web browser security research landed me internships and employment opportunities at Microsoft and Mozilla. After losing interest in security research, I got a job at Grooveshark where I wrote software to mine large amounts of user data, extract valuable pieces of information, and present them in a useful manner. During this time I played an integral role in developing the company’s analytics platform, Beluga, which aggregated music consumption patterns with demographic and survey data to glean insights on the audiences of different kinds of music.

While I have written and deployed several software projects, mainly web applications, the most complicated software contributions I have made have been to a product called Algebra Nation, of which I was the lead developer. While a software engineer at a consulting company called Wear Interactive, my team and I designed, built, and launched in three months a system for teaching remedial algebra in public schools. It integrated with Facebook and proprietary county school board login systems and could be used within a web browser or on one of the

Android and iOS mobile apps. Despite the breakneck timeline and constantly shifting client expectations, we were able to engineer a robust and relatively well-written system which supported half a million uses from day 1, and, while I am now longer with the company, the engineers there have had little difficulty integrating new features and scaling the system to two million teachers and students throughout Florida. Algebra Nation is hosted on the AWS cloud, and the server application, which can be deployed to an arbitrary number of frontend nodes behind a load-balancer, is written in PHP using the CodeIgniter framework. MySQL is used for the database, and all database operations use manually-written SQL calls optimized to make liberal use of database indices. On the client side, the browser application is written in spaghetti-style HTML/CSS/JQuery, and mobile apps are similarly written using PhoneGap, which allows for relatively straightforward cross-platform support by containing application code within a webbrowser object. Clients communicate with the server via a shared AJAX API.

My favorite language to work with is definitely Python. Its minimalist syntax and fairly consistent, predictable behavior make it a joy to use. The Python ecosystem contains a multitude of high-quality libraries for just about every conceivable need as well as a robust package manager for installing them. One very useful project, IPython Notebook, provides an interactive Mathematica-like shell in the web browser that makes it very easy to iteratively write code while examining its output in realtime. As someone coming from a PHP/Javascript background, Python is a real breath of fresh air.