Teaching Statement

Angelos Stavrou

My three-semester preceptorship as a PhD student at Columbia served as a strong motivation to pursuing an academic career. Being an instructor exposed me to teaching and advising students. This rare opportunity helped me realize that being a good teacher requires immense effort and passion. It is very challenging to keep students motivated and actively participating while making sure that they can understand the material.

This was especially true for introductory computer course I taught twice. This course, COMS W1001, “Introduction to Computer Science”, is geared toward non-majors. In this course, the thirty participating students had a very diverse educational background. A good teacher has to maintain a learning balance: instill excitement to the students that are more interested and motivated while making sure that none of the students are left behind. While most faculty can teach introductory classes, I think it takes a special effort to help students learn the material in introductory courses in a way that broadly defines the field (as opposed to being a glorified programming class). Introductory courses are often a student’s first impression of what the rest of the field is about. Since many students base their decision to explore a particular major on their initial experience from a few introductory courses, it is essential that introductory courses convey a representative overview of the field. We must enable students to find their interests, especially in an environment of relatively low CS enrollments. I believe in teaching a core set of principles and expanding from there: the underlying theme is important and stylistic constructs should come in only after the core material has been fully grasped.

To fully comprehend the material, I believe students should participate in laboratory sessions where they obtain hands-on knowledge and experience implementing what they have been taught during regular lectures. Fortunately, the experimental nature of Computer Science gives us a unique advantage over many other sciences: we can quickly implement and test our abstract designs on real, physical systems that are readily accessible to all students. In all the classes I have taught, both undergraduate and graduate, laboratory exercises were an inextricable part of the class curriculum. Student reaction was very positive: most of the students requested even more lab hours. In addition, all of the teaching assistants’ office hours were also held in the Lab.

At Columbia, I also had the unique opportunity to introduce and teach a new graduate-level course: Networking Laboratory (http://www1.cs.columbia.edu/ angel/netlab.html). The goal of the course was to bridge the gap between the theoretical Computer Networks Course and Internet technologies and protocols. Students had the opportunity to put ‘principles into practice’ in a hands-on environment using equipment usually available only to large Internet Service Providers (ISPs) including Cisco routers and end-systems. The curriculum of the course covered a wide range of systems topics related to networking and network security. The course was to designed to meet the following objectives:

- Comprehend via practical experimentation the fundamental design principles of Internet Protocols, IP addresses, and IP networks, including routing and forwarding.

- Implement and realize the problems and limitations of advanced Internet protocol technologies including network management, domain name system, network address translation, network management, and multicast.

- Apply their understanding of Internet protocols by analyzing, evaluating, and improving actual network configurations of IP routers and Internet enabled hosts in a controlled environment.
At the end of the semester, I wanted students to experiment with state of the art equipment used by next generation academic Internet (Internet2). To that end, Columbia hosted a four-day hands-on IPV4 and IPV6 Multicast Workshop at the Computer Science Department, where students were trained side-by-side with New York Internet2 operators using state-of-the-art Internet2 equipment.

The site http://oracle.seas.columbia.edu/index1.php contains student evaluations for the all classes I taught at Columbia University.

Being effective at teaching is a vital part of an academic lifestyle. It means being a leader both inside and outside formal classes. A teacher should be a role model to the students, inspiring them to learn while having fun at the same time. The class or the laboratory should be a learning environment where students are trained to think constructively and work effectively rather than obtain raw knowledge. Furthermore, a teacher has to guide the students to discover their interests, capitalize on their strengths and improve on their weaknesses.

Advising students is another aspect of being a good teacher. A good advisor should foster a creative and collaborative environment among students the allows them to develop their individual research ideas while having fun at the same time: research should be fun after all. The advisor should provide insights and help the students clarify their own ideas, ensuring that they work on interesting problems that will be well received by the rest of the academic community. Having frequent meetings with the students, especially at the beginning of their graduate life, is important. I have found this approach to be successful when I was mentoring Sambuddho Chakravarty, a Master’s student, with whom we designed a tool to measure Internet path characteristics. Sambuddho was successfully admitted to the PhD. program at Columbia where he continues to apply his research on the methods we developed to help solve other problems.

Based on my research and teaching experience, I would be excited to teach any security or systems related courses both in class and laboratory environments. Network security, overlay networks, security for large scale distributed systems, systems’ security and operating systems are some of the courses that I would gladly teach both as regular courses and/or seminars. Finally, I would like to design and teach new courses related to hot topics in security, intrusion detection and networks which would give students hands-on experience and a better research-level understanding of the current challenges of real world systems. More specifically, I would like to design the curriculum for the following classes: Security & Fault Tolerance for Internet Services and Operating Systems Security & Privacy.

Security & Fault Tolerance for Internet Services will cover the design challenges and security concerns of large-scale Internet services including routing and application-level security and modern types of attack against such systems. The goal of this class, which can be either graduate or undergraduate level, will be to combine a set of lectures with laboratory experiments, exposing students to the engineering requirements of large scale Internet services such as Voice over IP, Content Distribution and Thin Clients. The course will also cover current methods and best practices used to guarantee uninterruptible service and protection against a variety of network and application based attacks. Systems Security & Privacy will cover the fundamentals of operating systems security, giving an in-depth explanation on what prevents us from developing an overarching approach to dealing with system security and privacy. By implementing some of the modern attacks and protection methods, we will show that although there exist a lot of security mechanisms, systems are not currently designed with security & privacy in mind. Overall, both courses will focus on explaining why service availability and security on the Internet today depend on retroactive approaches rather than a well-thought plan.