# Teaching Statement

**Alireza Poshtkohi**
https://www.interdisciplinary.team

Developing new educational materials and techniques as well as teaching and working with students are my primary reasons for seeking an academic career. The essential components of an academic career are teaching and mentoring. I believe that teaching should be seen as a discipline in which the role of the professor adopts in response to the level, progress and feedback of the student(s) being taught. A teacher is first and foremost an educator. However, at times the teacher must act as a leader, mentor, a role model or even a peer for the benefit of the student.

## Philosophy of Teaching

I believe that learner-oriented teaching promotes learning that is both purposeful and enduring. As a teacher, it is my responsibility to know who my learners are, what kinds of knowledge and experience they bring to the group, and what they want to achieve so that I can tailor a curriculum that fits their needs and yet leaves enough room to accommodate topics that emerge from group discovery. My goal as a 'teacher of teachers' is to ignite in my learners a passion to create an institutional teaching and learning environment that fosters a conflagration of educational experimentation and innovation. My teaching philosophy views education in its many forms as an integral part of both personal and social development. Some students are curious and eager to learn who are easy to teach, but unfortunately, not all students in the classroom are in the same state of mind. To deal with this, I believe that engaging students in the learning process is the best solution in such a scenario. Students must be active participants in the learning process, rather than passive receivers and observers. Also providing students with a set of clear and realistic goals improve their thinking and their solving techniques. Challenging and achievable goals give the motivation for the students to respond in a better manner. Difficult assignments are frustrating and intimidating, while extremely easy assignments are boring, allow students to become careless, and do not give the students any sense of accomplishment. The following two quotes have inspired me throughout my teaching career:

"*The art of teaching is the art of assisting discovery*." –Mark Van Doren

"*A good teacher is like a candle – it consumes itself to light the way for others*." –Mustafa Kemal Atatürk

## Style of Teaching

My approach to teaching in the classroom addresses two fundamental educational strategies, including, active learning methods and interactive computing tools. Active learning methods ask students to engage in their learning by thinking, discussing, investigating, and creating. In class, students practice skills, solve problems, struggle with complex questions, make decisions, propose solutions, and explain ideas in their own words through writing and discussion. For example, in the activity of one minute paper, I ask my students to write a 1-2 min response to an open question used at the end of class through an online web form. I might ask "What is the most important thing they learned?"; "Summarize today's lecture in one sentence"; "What questions remain?"

I usually take advantage of interactive computing tools in delivering my classes, particularly Jupyter Notebook. The Jupyter Notebook is an incredibly powerful tool for interactively developing and presenting computing classes especially integrated with Python programming language. A notebook integrates code and its output into a single document that combines visualizations, narrative text, mathematical equations, and other rich media.

Before the first session of my class, I usually invite students to attend an induction meeting by which they can familiarise themselves with the module being taught, rules, teaching styles, etc. They will get essential information to help them succeed in the class. First, I talk to students individually and they speak about their general and specific interests in computer science, future career aspirations and so on. Here, I always try my best to encourage them towards computer science based on their answers. They also give a brief history of their educational track records, especially on skills that are vital to their success in the module. Then I focus on class rules and explain my expectations about acceptable behaviours within the class and the outcome of the class regarding the educational progress of the students. Finally, I provide students with two types of teaching from which they can choose one. Currently, there are two versions of choice: traditional classroom and blended classroom:

My traditional classrooms are highly interactive. The transfer of knowledge and information takes place from the teacher to the students, and any sort of query or difficulty flows from the students to the teacher. I pre-plan a module structure which is set for a traditional classroom. Regular tests are conducted to test the understanding of the students of the modules. Various classroom activities like debates, group projects, and group discussions are conducted in my traditional classrooms.

In the Blended learning approach, I combine online educational materials and opportunities for interaction online with physical place-based classroom methods. Especially, I can take advantage of flipped classrooms that invert the traditional learning experience. Lectures are pre-recorded and shared outside of class time for individual review as homework, and classroom time is reserved for class discussion, interactive projects and improving student aptitudes in problem-solving. By shifting passive lecture material to an at-home setting, students can review those materials in the time and place that works best for their needs. They can also access all the foundational information in advance, so when they enter the classroom, they feel prepared and ready to participate in interactive learning activities.

I always incorporate inclusive module design into my teaching ranging from accessible learning resources (such as captioned videos, multiple formats for instructional material, etc.), inclusive language (such as gender-neutral words, jargon-free teaching material, etc.), culturally relevant examples (such as using real-world problems encountered by diverse cultural backgrounds) and flexible assessment strategies (such as allowing the students to choose their favourite type of assessments through projects, presentations, written assessments or exams).

After my classes, I often request the students to evaluate my teaching and the class atmosphere and then give me written or verbal feedback. This helps me adapt the module materials as well as my teaching style effectively throughout the semester. At the end of every semester, there is another anonymously written feedback on the pros and cons of my classes which are used for my next academic year.

My teaching is usually influenced by the latest research in the literature. For example, I have benefited from parts of my book in most of the modules that I have taught. That book is accompanied by the latest research or my own research efforts in HPC software engineering. Part of my teaching involves students doing a number of small-scale research activities to enhance their research-based learning skills.

My previous experience showed that by using the approaches above the overall satisfaction of students regarding their progress and educational achievement for a specific module is maximised.

## Previous Teaching Experience

My first exposure to teaching was at high school when our maths teacher was ill for a while, and I taught on their behalf. I have worked in academia since then. I have designed and taught many different undergraduate/postgraduate-level modules in computer science and electrical engineering because I always adore teaching and exploring new topics in depth. I managed to inspire non-CS students (e.g., communication engineering and dentistry) to become interested in computer programming and CS fundamentals to improve their way of thinking and creativity. Please note that in most of the modules below, I have benefited from parts of my book.

In **cloud computing**, students delve into cloud architecture, service models, and practical deployment. By exploring diverse platforms like Azure, AWS, and OpenStack, students gain a comprehensive understanding of cloud technologies. Emphasising problem-based and reflective learning, the module equips students with analytical skills to assess and implement cloud solutions for business needs. The coursework, comprising research and practical activities, ensures a holistic evaluation of cloud computing concepts. The module's structured weekly plan, incorporating labs and lectures, fosters hands-on experience and facilitates a seamless integration of theoretical knowledge and practical skills. Through continuous improvement and real-world applications, this module aims to empower students with the expertise needed in the rapidly evolving field of cloud computing.

In **parallel algorithms**, the students get familiar with the foundation of parallel computers from hardware, OS and parallel programming languages (such as MPI, OpenMP, Intel TBB, Cilk, CUDA, etc.) to synchronisation algorithms, parallelisation approaches to sequential problems and Cloud Computing. Final term projects are within the context of HPC, parallel scientific computing, parallel computer architecture, acceleration of cryptographic operations on GPUs and FPGAs, the energy efficiency of parallel algorithms and data centres, and concurrent and distributed systems. Each project is carried out by a group of at most five students.

In **Parallel and Distributed Simulation**, and **Performance Evaluation of Computer Systems**, the students are concerned with learning the basics of queuing theory, discrete event simulation (DES), parallel DES (PDES), priority queues, and performance evaluation of parallel systems and computer networks. Each student must describe, code and simulate the model of a small-scale discrete-event problem in computer science (such as vehicular traffic, airport, logic circuits and computer networks) in a C++-based library for DES simulation that I had implemented during my research on PDES. Like other classes, they are introduced to a topic to carry out research and present their results at the end of the semester. Previous projects, to name a few, were concurrent priority queues, the history of simulation languages, static and dynamic code instrumentation, and parallel debugging support for MPI.

In **computer networks** and **operating systems** courses, I teach in a way that the students are motivated to put their hands practically in programming and implement some not-very complex but yet important concepts, including, network and systems programming.

In **logic circuits**, **computer architecture** and **hardware/software co-design**, I prepare the students to work with different simulators and hardware platforms (SoC and NoC) and do their course project by implementing an interesting and demanding problem with a complete set of hardware and software stacks and possibly network. A number of sample projects were: implementing an embedded web server connected through an Ethernet network card directly to a LAN socket, adding a customized instruction or unit to a pipelined MIPS processor, working with HLS tools and full system simulation using virtual platforms run by SystemC.

In **advanced programming** modules (C, C++ and C#), I managed to promote the student's level of thinking and came up with different solutions for various given problems, by providing them with examples, homework and classroom quizzes. I carried out the learning process of the programming languages by helping the student become familiar with the syntax and semantics of the programming languages that they were studying. My **web development** course is mainly presented in totally two different variants: PHP/MySQL and ASP.NET/MSSQL[1]. I give a collection of website templates to students, and they must customise them and add a module through web programming. They are supposed to learn CSS/HTML/JavaScript within the first three weeks of the semester.

In **computer networks and OS labs**, I acquaint the students with the necessary hardware and software stacks of OS design and network OSI layers. For the former lab, the students are provided with Linux kernel modules and system call development, then they are divided into different groups, each of which should implement a project related to driver programming or enhance a part of the Linux kernel (such as either processor or disk scheduler). In the latter, the students learn practically different aspects from network hardware (e.g., routers and switches), network design, simulation (in NS2 or OMNeT++), data centre principles, Cloud computing, and network programming to network administration. Both Linux and Windows network-based variants are discussed.

In **advanced algorithms**, I present the basics of parallelisation algorithms and machine learning, advanced graph algorithms (e.g., graph partitioning), divide and conquer algorithms, etc. The class is arranged into four or five groups to do research on a specific algorithmic area by reading research papers or textbooks, for instance, parallel graph traversal, evolutionary algorithms, artificial intelligence, and genetic programming. They are required to present their findings at the end of the semester in a conference hall. Through this course, I encouraged the students to read recent publications and make sure they understand the problems studied in these publications and discuss future ideas for improving previous works or proposing new solutions.

In the **circuit theory** module, besides several collections of exercises, the students must verify their analysis with circuit simulation software like SPICE. If we have enough time during the semester, I will also present advanced topics related to non-linear circuit theory, filters & circuit synthesis, algorithms used in the design of CAD tools, and communication circuits (for whom it will not likely be as a course in the future).

In **BSc/MSc supervision**, I usually benefit from a scaffolded dialogue by which students receive support and guidance as they learn new skills or concepts. My goal is to gradually withdraw the scaffolding support as my students build more background until they become completely independent.

---

[1] https://github.com/poshtkohi/web-development