Statement of Teaching

H.M.N. Dilum Bandara

The exhilaration I experienced when I first set foot on the stage for my first class in late 2004, continues to persist undiminished, even close to *two decades* later. I have gained extensive experience in tertiary education across three countries, effectively delivering over 40 modules, designing over 80 syllabi, and developing four curricula. Starting as a Lecturer at the University of Moratuwa (UoM), Sri Lanka, I later served as a Graduate Research/Teaching Assistant and a Substitute Lecturer at the Colorado State University, USA, and mostly recently as an Adjunct Senior Lecturer at the University of New South Wales (UNSW), Australia. Moreover, I have taught five modules adhering to the academic standards of the University of Greenwich and the University of the West of England, UK. Furthermore, I participated in a 3-week seminar on the Chinese Education System held in China.

Teaching Philosophy

I view teaching as a blend of art and science, requiring continual mastery due to the constantly evolving student expectations and the Computer Science and Engineering (CSE) discipline. Recognising that not all students approach a class with equal motivation—many tend to view an undergraduate CSE degree as a stepping stone rather than a goal in itself—my primary focus lies in creating an environment that *fosters and nurtures each student's latent desire to learn.* I motivate students to take charge of their personal and professional growth. While exam preparation holds importance, I prioritise a solid understanding of fundamentals, fostering hands-on skills (e.g., software design, development, and debugging), and soft skills. I advocate for *assessment for learning* over the assessment of learning. I ensure that lectures, labs, projects, and exams are engaging, suitably challenging, aligned with each student's interests, and constructively aligned. Group projects, which allow me to interact more closely with students and embrace their diverse cultural, skill, attitude, and technical backgrounds, bring me immense satisfaction, particularly when struggling groups ultimately rise to the occasion and deliver their finest work. Following the university's role of credentialing students, I also better prepare my students for exams by sharing tips on exam skills. Recognising my moral duty to foster engineers who possess inquisitive minds, ethical principles, social responsibility, and an awareness of the impact of their actions, I incorporate these discussions into my classes as appropriate.

Teaching Style

Top-down delivery and bottom-up assessment. The technology students encounter daily surpasses the basics taught in class by miles, leading them to lose interest in foundational concepts and gravitate towards abstract ideas. Even though recent textbooks and corresponding classes typically adopt a bottom-up approach, painfully covering all basics before putting things together to show the big picture, *my classes have proven more effective using a top-down strategy.* Beginning with recognising the latest developments, I gradually deconstruct complex topics, focusing extensively on one basic concept at a time. This emphasis on fundamentals is crucial, as it is the only invariant, not the jobs students will undertake nor the tools to do those jobs. Generative AI offers a new opportunity to transition from intricate topics to basics, reducing friction in learning and student inertia. Properly designed and facilitated, it enables students to swiftly generate multiple solutions, dedicating more time for analytical reasoning about their code quality, style, and software qualities; algorithmic thinking; and problem-solving. However, *my labs follow the bottom-up approach*, mirroring real-world construction processes. This mix gives students an appreciation of the basics and their connections to higher-level abstractions. This teaching method has demonstrated effectiveness in conveying engineering principles such as abstraction, hierarchical design, and component-based construction in modules like first-year Computer Architecture (UoM) and Software Architecture for Blockchain Applications (UNSW). Similarly, it has been successful in my Digital Finance Technology module, delivered to a cohort of finance, law, and computer science PhD students from five Digital Finance CRC (DFCRC) partner universities, earning recognition with Data61's Customer First award.

Match design, delivery, and assignments. In crafting daily lesson plans, *I prioritise quality over quantity*, considering foreseeable industry and academic relevance, student comprehension, expectations, and learning outcomes. For instance, since undertaking the leadership of the Software

Architecture for Blockchain Applications (UNSW) module, I streamlined content to prevent rote learning, recognising that many students lack the foundational software engineering understanding to appreciate software architecture design. I also introduced new topics like requirements analysis to bridge this knowledge gap. My large, lower-level undergraduate classes emphasise skill development, real-world relevance, and preparation for advanced courses, while smaller, higher-level classes focus on fundamental understanding, analysis, and soft skills. Thus, my introductory modules feature lectures, labs, quizzes, and field visits, while advanced modules prioritise student presentations, discussions, games, and group projects. A low-stake or high-stake writing component is integral to most of my classes, fostering writing across the curriculum to enhance graduate competency. I encourage and reward in-class participation and derive exam questions from student-led discussions beyond textbook/slide content, a practice disclosed to students. This constructive alignment has resulted in consistently high attendance, excellent grades, and positive feedback in my classes. A UNSW student noted, "*This is the only course that provided me an end-to-end exposure to software engineering, from drafting real-world requirements to architecture design, development, and testing*". Another credited their job success as a software engineer to knowledge gained in my class.

Engage the students. I strive to align lectures, labs, group projects, and assignments with students' interests and abilities to maintain their engagement inside and outside the classroom. For example, I often *empower students to propose project ideas within specified guidelines*, fostering a sense of ownership and allowing them to select a team that matches their interests and complements their skill sets. This approach ensures that even weaker students can contribute and receive respect and recognition. During project meetings, while weaker students may remain quiet or piggyback on others, I actively engage with students by circulating the tables, providing them with the opportunity to share their progress and blockers. *On-the-fly assessments*, such as spontaneous questions, demonstrations, and quizzes, have proven effective in increasing students' level of concern. Such assessments and nonverbal feedback from students help me to adjust the pacing of classes accordingly. Recognising the significance of enterprise social media tools in the industry, I incorporate Wikis, forums, and chat groups into my modules. These tools also help me alleviate interaction and collaboration barriers, especially in online, large lecture-based, or classes with junior students. I maintain an open-door policy and cultivate a friendly yet professional rapport with students (sometimes I initiate one-on-one meetings), as I understand that these small gestures ultimately influence students' overall reactions towards the class and learning in general.

Storytelling. An engaging and succinct narrative about the module is crucial for inspiring and stimulating the students. Presented during the first class, my narrative elucidates the relevance of class content in the industry, outlining the skills they will acquire to pursue their career goals and my assessments to measure those skills. Additionally, I define success at the semester's end and delineate the path to achieving it by meeting my expectations and adhering to acceptable conduct. Throughout the semester, I revisit various elements of the narrative to reinforce key points, often drawing from my research and industry encounters as illustrations.

Curriculum Development

As the CSE department's industry liaison at UoM, *I engaged in active discussions regarding industry needs*. Consequently, in 2014, I introduced the Business Analytics specialisation and portfolio option to the MBA in IT program. Also, I arranged a field visit and industry experts to deliver guest lectures. These efforts doubled program intake and on-time graduation during my tenure. In 2022, I developed the Digital Finance Technology module for DFCRC, which is now being adapted into the Digital Finance MBA to be launched across partner universities. It was based on the Software Architecture for Blockchain Applications (UNSW) module, incorporating about 40% new content and a new delivery strategy to address the intersection of technology with finance and law. The module facilitated experimentation with immediate feedback from students and peers. I applied the insights gained to substantially revise the content of DFCRC and UNSW modules closing the loop. This year, I consulted the South Eastern University of Sri Lanka to *develop a new BSc Hons in Computer Science curriculum* with the ACM/IEEE-CS/AAAI Computer Science Curricula 2023 and SLQF standards. Further, I coordinated the establishment of a Partner Program, two industry-sponsored labs, a student workspace, and facilitated the signing of an MoU with the University of Seychelles.