

# Teaching Statement

Michael Fu

yeh.fu@monash.edu

<https://michaelfu1998-create.github.io/>

I am excited to mentor, advise, and teach students as part of my faculty role. My approach to teaching and mentoring is grounded in core principles that I have identified throughout my teaching, research, and mentoring experience.

## Teaching Experience

**Software Engineering.** In 2024, I was invited as a guest lecturer for *FIT5136-Software Engineering* at Monash University, delivering a postgraduate-level online lecture on AI for DevSecOps. My role involved introducing the DevOps workflow and emphasizing how AI technology can enhance security throughout the DevOps lifecycle. The student evaluations highlighted my engaging and enthusiastic teaching style, with feedback praising the lecture’s clarity, informativeness, and practical relevance.

**Software Engineering.** In 2024, I was the Teaching Associate (TA) for *FIT5136-Software Engineering* at Monash University. This postgraduate unit provides an introduction to the discipline of software engineering, covering various aspects such as the software life cycle, programming by contract, design and testing issues, and project management. As a TA, I delivered weekly applied sessions to a group of 24 students, assisted with group collaboration issues, provided online consultations, and handled assignment and assessment marking.

**Deep Learning.** In 2024, I was the Teaching Associate (TA) for *FIT5215 - Deep Learning* at Monash University, a postgraduate-level unit focused on modern machine learning and deep learning techniques for data science and AI applications. My role involved delivering lab sessions weekly to groups of 60 students, where I introduced the technical details of deep learning models, guided students through coding for building and training models, and assisted with hardware and network issues related to model training and cloud service usage. I also hosted in-class Kaggle competitions to enhance student engagement, provided online consultations, and handled assignment and assessment marking.

## Teaching Philosophy

My teaching philosophy has been shaped by my own experiences over the eight years I have spent in higher education starting from my bachelor’s study, as well as by the great teachers I had the fortune to come across.

**Mastering Fundamentals & Embracing Versatility.** I believe that teaching should equip students with the ability to solve a wide range of problems by understanding the solutions that have already been discovered. As Richard Feynman said, “*Know how to solve every problem that has been solved.*” In my teaching experience in *FIT5136* and *FIT5215*, I emphasize mastery of fundamental concepts, allowing students to apply proven methods to new challenges. However, I also recognize the importance of diverse approaches, as highlighted by Charlie Munger’s warning that “*To the man with only a hammer, every problem looks like a nail.*” To address this, I encourage students to explore multiple problem-solving strategies. For example, in *FIT5136*, I guide students through various software design patterns, showing them how different techniques can be applied to similar problems, thus broadening their problem-solving toolkit. By combining a strong foundation with an openness to diverse approaches, I aim to prepare my students to tackle both familiar and novel challenges with confidence and creativity.

**Being Passionate & Inclusive.** As a teacher, I recognize the inherent asymmetry of passion between myself and my students. While I may have taught the same course multiple times, my students are often encountering the material for the first time. To bridge this gap, I believe it is crucial to bring genuine enthusiasm to my teaching. By maintaining a high level of passion and responsiveness to student inquiries, I aim to create an engaging learning environment where students feel motivated and supported. Furthermore, I am committed to inclusivity, adapting my teaching to meet the diverse backgrounds and expertise levels of my students. I share advanced topics and extended content with those who have prior knowledge, while patiently answering inquiries and encouraging students with no background. Aware of challenges like language barriers, I draw on my experience as a non-native speaker with a speech impediment to create an inclusive environment. My strategies include offering multiple participation avenues, providing extra support, and being approachable and patient.

## Course Plans

Given my background in software engineering, software security, and deep learning, I would be happy to teach classes related to *software engineering/security, algorithms and programming foundations in Python/Java, deep learning, machine learning, and natural language processing* at both postgraduate and undergraduate levels.

For example, for the introductory undergraduate-level software engineering class, I will focus on DevOps by teaching students the *fundamentals of continuous integration and deployment, version control with Git, and the basics of containerization using Docker*. This approach will give students practical skills in automating development workflows and managing modern software systems. For the advanced postgraduate-level software engineering class, I will integrate security principles into DevOps practices to emphasize DevSecOps. As we explore advanced DevOps techniques such as continuous integration and deployment, containerization, and infrastructure as code, *students will also learn how to incorporate security measures throughout these processes*. This approach ensures that future developers are not only proficient in modern software engineering practices but also equipped with the necessary security knowledge to build resilient and secure applications in today's threat landscape.

## Mentoring

**Mentoring experience.** Throughout my career, I have had the privilege of mentoring several students across various projects. Notably, I mentored Yuki Kume, an undergraduate student at Monash University, through a research project on an AI-driven cybersecurity tool. Together, we transformed our research idea into a practical AI-driven security analysis tool integrated into VSCode, which has garnered nearly 1,000 downloads to date. Currently, I am guiding three undergraduate students at Monash University on their final year project, which focuses on developing a no-code citizen developer chatbot for training machine learning models. This project aims to make machine learning more accessible to non-technical users by simplifying the model training process. Additionally, I mentored Rui Yang, a junior Ph.D. student at Monash University, alongside my supervisor, in an industry collaboration project centered on AI safety. Rui is working on implementing safety measures in AI systems to address ethical and operational challenges. I also provide informal mentorship to junior Ph.D. students in our lab, offering feedback on their research ideas and general advice about navigating the Ph.D. process.

**Mentoring philosophy.** My mentoring philosophy is centered on fostering independent thinking and confidence in my students. I aim to guide them in developing their research questions and becoming confident in their ideas. I strive to learn from my students as much as I teach them, encouraging them to explore their passions and tailor my support to their individual needs. I'm also committed to providing research opportunities for undergraduates and offering extra support to students with disabilities and those from underrepresented backgrounds.