# **Teaching Statement**

#### Muhammad Zubair Malik

I am passionate about teaching. My inspiration comes from Richard Feynman who left us this final message on his blackboard: "*Know how to solve every problem that has been solved*". Formal learning gives us the means to stand on the shoulders of giants. We get the opportunity to learn the ground breaking works that have stood the test of time and shaped our world. My teaching philosophy is to focus on the fundamentals and help my students develop an appreciation for the problem. I foster their skills to exploit existing solutions with ease and achieve mastery. I tickle their curiosity and inspire them to explore further with scientific objectivity and courage to persevere.

## **Teaching Experience**

I have worked as a head TA for Introduction to Programming using C, and Software Design and Implementation I courses at the University of Texas at Austin from 2009 to 2012. These courses were offered every semester (including summers), and were mandatory requirement for electrical engineering, software engineering and biomedical engineering students. Students from computer science, mathematics and other engineering departments also took it in large numbers. We introduced the students to the fundamentals of software development and programming including the C language (and a little bit of C++), algorithm design and analysis, and software validation through testing. These were large classes with many pedagogical as well as managerial challenges. Most significantly, the students of different capabilities and varied backgrounds tested my ability to communicate complex ideas effectively. I have been involved in all aspects of the course i.e., leading the recitation groups, lecturing as replacement instructor, designing and grading assignments have rated me excellent in enthusiasm, punctuality, knowledgeability, communication abilities, openness to questions, respect and helpfulness. My average for all the courses is 4.5/5.0, which is way above the department average. Other undergraduate courses that I have assisted in teaching at UT Austin include Data Structures and Algorithms, and Software Testing.

At the graduate level, I have worked as a TA for Engineering Programming Languages, Advanced Compiler Techniques, and Software Verification and Validation. These courses are at the core of software engineering curriculum and build the necessary skills for advanced scientific and industrial work. In these courses I have helped in grading, lecturing, designing exam questions and providing support to the students.

In 2008, I worked as a teaching fellow at the Lahore University of Management Sciences (LUMS). This post is equivalent to co-instructor and I was involved in teaching of *Speech Processing*, *Assembly Language Programming* and *Advanced Programming Techniques*. Prior to this I had worked as a TA for *Advanced Programming Techniques* and, *Circuits and Systems* at LUMS. I have also worked at National University of Computer and Emerging Science (NUCES) from 2002-2003, where I was TA for *Digital Signal Processing* and *Computer Networks*.

### **Research Mentoring**

Working as a Research Fellow (2013-2014) at the Center for Identity, I lead the development effort for Identity EcoSystem Project under the supervision of Dr. Suzanne Barber. I mentored two outstanding graduate students, Shiyani Debb and Liang Zhu — concretizing their research goals set by Dr. Barber, exploring possible design hypotheses with them, defining deliverables, setting deadlines, and debugging and integrating their code. Our project was demonstrated live at ID360, an invitation only conference attended by companies like IBM and agencies like United States Secret Service. Our work was highly well received by the participants.

I advised and collaborated with a graduate student, Megan Ruthven, during summer and fall of 2012 on automated program repair under the supervision of Prof. Dewayne Perry at the Center for Advanced Research in Software Engineering. I worked on clarifying the research questions, research methodology and helped to implement technically challenging aspects of her work. That work was published as a report and will be submitted to a conference.

I worked as a research officer at Multimedia Lab at NUCES, where I lead the H.323 Reference Implementation project and advised a team of 8 undergraduate students. The undergraduates required more specific guidance and I had to specify each of their parts very precisely. I also had to write tests in advance to make the specifications clear for them. In the capacity of research officer at NUCES, I was also an adviser to many undergraduate final year projects. I was responsible for providing data analysis support, implementing and enforcing project report templates, and providing project management support to project supervisors.

#### **Teaching Interests**

I am interested in teaching courses in Software Engineering. At the undergraduate level, I am well equipped to teach Introduction to Programming, Empirical Software Engineering, Introduction to Algorithms, Data Structures, Software Testing, Programming Languages and Compiler Design. I put a lot of effort in preparing structured material for my undergraduate students. I start with concrete examples, collaboratively solve problem instances with students and then generalize and present the formal theory. Building instructive examples that can highlight take-home messages is a hard work but is always appreciated by the students. I try to challenge the brilliant students of the class to maintain their interest; I ensure that everyone develops the critical skills; and I prepare alternative and easy to understand examples for struggling students.

At the graduate level, I will be excited to teach Software Verification and Validation, and Program Analysis and Synthesis. It is becoming increasingly important to learn and reason about program correctness in a formal way along with the use of traditional approaches. Tools that can reason about behaviors of programs are becoming more and more powerful and practically useful. I would like to cover the fundamental techniques that have enabled automated bug finding to improve software quality, formal verification to guarantee absence of bugs in critical software, advances in languages that enhance security and modularity, and optimization techniques.

For the sake of brevity, I have highlighted only some of my interests. I am not limited by these and am always happy to consider interesting possibilities. Teaching is not only an act of imparting knowledge, it also entails training students to learn on their own and explore beyond the classroom. I feel that the depth and diversity of my experience has better prepared me to help my students achieve both their personal and academic goals.