Pedagogisk mappe– Muhammad Rizwan Ali

Background

I'm currently working as a Ph.D. candidate at Western Norway University of Applied Sciences (HVL), Bergen, Norway. I have an extensive teaching experience of approximately eight years. I have taught different courses at the bachelor's level. Moreover, I also have been a team leader for different programming laboratory courses. I am planning to record a series of short lectures to benefit the students outside the classroom.

I have experience in smoothly executing all the stages of a course. As per my understanding, a course may go through the stages as follows:

- Initiating and creating a new course from scratch.
- Deciding/designing and developing a pedagogical methodology or a strategy for the course.
- Preparation and delivery of lectures with the concepts aligned timeline
- Reinforcing, testing, and evaluating the learned knowledge through different tools such as quizzes, assignments, group activities, laboratories, and projects.

I have a broad knowledge of all these stages and have gone through them iteratively. I can develop all the required instruments for a course within the given time frame.

Education

Education in the Subject Area

I am in the final year of Ph.D. in Computer Science at Western Norway University of Applied Sciences, Bergen, Norway. Earlier, I have earnt a Master of Science in Computer Science from COMSATS Institute of Technology (CIIT), Lahore, Pakistan, in 2018 and a Bachelor of Science in Computer Science from Punjab University College of Information Technology (PUCIT), Lahore, Pakistan, in 2014.

Pedagogical Training

To hone my pedagogical skills, I have attended a fifteen-day full-time intensive pedagogy training at the University of Central Punjab (UCP), Lahore, Pakistan. The training theme was to improve teachers' existing pedagogic practices. During this training, we were introduced to different classroom management theories, such as choice theory by William Glasser, general psychology and educational psychology. For the practical part, the classroom environment was simulated. Moreover, different pedagogic activities such as body gestures during the class, how to deliver a lecture and efficiently use a whiteboard, engaging students to keep their interest alive and strategies to improve the classroom environment were discussed and practised.

Subject Area Knowledge

I started programming before starting my bachelor's education. After completing my bachelor, I joined academia; my professional background helped me to deliver concepts with real-world examples and related explanations. I also kept myself updated about the latest technologies by remaining in touch with people in the industry. Later, the master's degree and research during my Ph.D. broadened my knowledge, reflected in my teaching. I also kept myself updated with the latest standards and changes in development tools and languages.

Teaching Experience

I have taught a total of approximately seven different courses in two different universities in two different countries, i.e., Pakistan and Norway. The name of the universities and my teaching duration in these universities are as follows: University of Central Punjab, Lahore, Pakistan (2014-2019) and Western Norway University of Applied Sciences (HVL), Bergen, Norway (2019 - up to date). I have served as a permanent faculty member in the capacity of lab teacher (2014-2017), associate lecturer (2017-2018) and lecturer (2018-2019) at UCP. While at HVL, I have assisted the teachers as a Ph.D. candidate.

Table 1 provides the basic information about courses, such as their level and number of students. Table 2 presents the detail of my role and involvement in each subject. It is worth noting that Table 2 also indicates my steady growth in terms of course responsibilities and involvement. For most of these courses, I have been involved in everything from lecturing to student evaluation. Out of these seven courses, there are three courses such as Introduction to Programming, Data Structures and Algorithms and Object-Oriented Analysis and Design, which have been either newly developed or revised to be new courses. For other courses, lecture contents are improved, and new assignments and projects are developed. A selection of course material is available online [1].

[1] https://github.com/razi236/Pedagogisk-mappe

Table 1: Details of subjects taught and their level

	Subject	**Level**	**Times**	**Avg No of Students**
UCP5	Object-Oriented Analysis and Design	Bachelor	5	40-50
UCP4	Data Structure and Algorithms	Bachelor	7	40-50
UCP3	Object-Oriented Programming	Bachelor	8	40-50
UCP2	Programming Fundamentals	Bachelor	2	40-50
UCP1	Introduction to Programming	Bachelor	5	40-50
HVL2	Videregående webapplikasjoner	Bachelor	1	20+
HVL1	Mobile og Distribuerte	Bachelor	1	20+

Table 2: Role and involvement in every subject

Subject	Year	Role					
		Lecture	Lab	Material Development	Evaluators	Examiner	Team Leader
UCP5	2018-2019	✓	✓	✓	✓	✓	✓
UCP4	2016-2018	✓	✓	✓	✓	✓	
UCP3	2014-2018	✓	✓	✓	✓	✓	
UCP2	2016	✓	✓	✓	✓	✓	
UCP1	2014-2019	✓	✓	✓	✓	✓	✓
HVL2	2020		✓		✓		
HVL1	2019		✓		✓		

UCP University:

After completing my bachelor, I joined UCP as a lab teacher. During my first year, I was responsible for preparing and conducting the labs and preparing the evaluation instruments. My participation in the instruments' preparation was approximately 50%, and I was under the supervision of a senior colleague. During this period, I followed a traditional teacher-led classroom methodology. I independently created and corrected the exams and evaluated the students. This period helped me to learn class management, and interaction with students and helped me to realize that as a teacher, I need to help students in their learning. Moreover, I found that most of the time, the communication was from one side and students were passive. It was hard to determine whether they are learning or not.

After my promotion as an Associate Lecturer, I have taught programming courses, i.e., Introduction to Programming, Object-Oriented Programming, Data Structures and Algorithm and Object-Oriented Analysis and Design. Also, I worked as a leader of the lab teachers and helped them prepare programming labs, assignments, and quizzes. I consider that time a transition period from a teacher-led classroom approach to a student-oriented approach. All the courses taught were converted to problem-based learning and achievable intended learning outcomes were written for each course.

Introduction to Programming:

This course was re-designed and divided into three parts. The first part aimed to develop the logical thinking of students. Thus, students were given logical problems, which can be solved using common sense, general knowledge, basic mathematics and a pen and paper. In the second part, programming concepts were introduced indirectly by creating

animations in the Scratch software[2]. In the third part, students were taught C++ syntax, and programming concepts learned in Scratch software were mapped with C++. Each lecture was divided into two parts, i.e., theory, mapping of scratch concept and problem solution. Thus, a theoretical part was taught in each lecture and multiple related problems were given to solve. The problems were categorized into two categories, i.e. basic and intermediate/advanced.

For the basic problems, the students were suggested to work in small groups to find a solution. The merits and demerits of solutions presented by students were discussed. Later, a combined or better solution was developed by interacting with students. Afterwards, students were required to work individually on the remaining problems within the class. Students' progress was continuously monitored and required help was provided during the session. Homework could have been given at the end of the lecture or on a specific topic. A weekly assignment was also part of the course.

Moreover, the concept of students learning from students was introduced in this course. For this purpose, the top students of the preceding semester were selected as volunteers. These students were present during the class; their job was to identify students who were less active in the class and help them during the practical work. The hypothesis behind this practice was that students might be more comfortable discussing their problems with students than with teaching staff.

Programming Fundamentals:

This is the second course for the programming stream. During this course, I helped the coordinator (course team leader) convert the course to a project-based learning course. This course followed a partial teacher-lead classroom strategy for the theoretical part to develop the content knowledge and explorations of real-world problems. In the laboratory part, students worked on mini projects to develop the solution for the problem and a larger project later, reinforcing all learnt concepts.

Object-Oriented Programming:

When I taught this course the first time, I used and followed the existing course contents and material. However, I redesigned the course contents from the second occurrence and designed new evaluation instruments and a different teaching methodology. The course became more students centric when I focused on implementing real-world problems. Also, the students were free to think about some real-world scenarios and map them with the programming concepts learned in this course. This methodology increased the student's interest in this course, and they implemented outstanding projects. It is worth noting that this course was offered using both C++ and Java languages.

Data Structure and Algorithms:

This course was very challenging in terms of teaching and re-designing. The students enrolled in this course had different programming skill levels, i.e., from basic to comfortable levels. One of the purposes of this course was to bring all the students to an acceptable level. For this purpose, the course contents were re-designed and weekly laboratory sessions were introduced. A real-world problem was selected as a main theme over the course. Each concept in class was directly related to a particular part of the theme. The lecture was divided into three parts, i.e., basic theory, introducing the relevant part of the theme and associated problem, and applying the theory to write the solution to the problem. After the weekly lectures, students were required to work in the laboratory and implement the learned concepts. The class teacher, with at least two teaching staff, was at the laboratory to provide guidance. Students were encouraged to do experiments with different theoretical concepts and bring novelty to their solutions.

Further, they were required to reflect on their solution. A project was also given to students, which was evaluated based on their level of learning and applying the concepts and their reflection on their project. It is worth noting that this course was offered using both C++ and Java languages.

Object-Oriented Analysis and Design

This course was a replica and merger of two courses, i.e., Object-Oriented Programming and Design Patterns. However, there were different challenges involved in this course. The first challenge was to merge the contents. The second challenge was to deliver the contents of two semesters in one semester without compromising the quality. The third challenge was to keep students interested. The fourth challenge was learning with students and bringing them to an equal level as those students who learned these concepts in two semesters. The fifth challenge was to redesign all the evaluation instruments. These challenges were addressed by defining a suitable teaching approach and methodology. I taught the theoretical concepts with examples. Later, students were required to sit in groups and solve a problem. Once

[2] https://scratch.mit.edu/

the whole class solved the problem, the next theoretical concept was introduced, and the steps were repeated. At the end of class, a part of the project was given to solve.

HVL University:

At HVL, I was involved in two different courses. Both courses followed the project-based learning methodology and had clearly defined intended learning outcomes (ILOs).

Mobile og distribuerte applikasjoner

For the mobile og distribuerte course, I was responsible for introducing the lab, helping the students and evaluating the students' learning. This practice gave me global experience and familiarized me with the pedagogical approaches, mostly student-oriented, which are practised in Norway.

Videregående webapplikasjoner

For the Videregående webapplikasjoner course, I was mainly responsible for helping students in labs and evaluating the practical work, such as programming assignments and projects.

Digital Learning

At UCP, I have also worked in the Centre for Learning Design to transform the educational landscape of Pakistan and make education more effective, accessible and affordable for all students. We worked in collaboration with the Centre for Game Design and the Centre for Robotics & Security to radically change the field of education and revolutionize the methods of imparting education. Moreover, we tried to facilitate the students at all levels of educational attainment with special attention paid to developing affordable educational tools and applications using m-learning concepts for students from under-serviced communities. At HVL, during the Covid, I conducted the labs digitally. Several advantages were noted, including a dramatic increase in students' learning, flexibility, accessibility, and cost-effectiveness.

Supervision Experience

I have been extensively involved in the supervision of bachelor final year projects. A list of these projects is given in Table 3. I'm capable of supervising multiple projects at the same time. Each project was a year-long and followed software engineering guidelines from requirement engineering to product delivery to simulate the real world.

My style of supervision is very flexible and supportive and contains a close collaboration with students. I focus on students and their problems. I discuss their suggested solutions for the problems and provide constructive feedback. I also provide continuous feedback on each milestone to improve the quality. I motivate students to explore new avenues for finding a solution. I also help them to progress and become independent gradually. I also help them to break down the problem into realistic and achievable goals.

Table 3: A list of bachelor projects and Master theses.

Sr. No	Project / Thesis title	Institution	Level	Status	Year
1	Formal Modelling of Aircraft Maintenance System	UCP	Bachelor	Completed	2019
2	IoT Based Irrigation and Fertilization of Crops	UCP	Bachelor	Completed	2019
3	Formal Modeling and Validation of Smart Train System	UCP	Bachelor	Completed	2018
4	Formal Modeling and Verification for Balanced Load in On-Demand Computing	UCP	Bachelor	Completed	2018

Management in the field of education

In UCP, a particular course is offered to multiple sections simultaneously. Thus, a team of teachers normally delivers these lectures. A team leader is appointed. I have been appointed as a team leader for the Introduction to Programming and Object-Oriented Analysis and Design lab course. I have been responsible for designing and aligning the lab contents with pre-requisite courses and courses offered after the particular course with help from the team. As a team leader, I also have been responsible for the smooth execution of the course and ensuring that all the sections are at the same level at a specific point in time. I also have been responsible for approving instruments developed by different teachers and producing a joint exam for all the students. A weekly meeting was held to discuss the problems raised during the last week, and necessary actions such as repeating or reinforcing concepts were decided. As a team leader, I changed the course from a teaching-lead classroom to problem-based learning and introduced constructive alignment. The colleague teachers were encouraged to write down clear intended learning outcomes and develop a strategy to achieve them. I also encouraged my colleagues to borrow real-world problems and scale them down to the classroom level. Moreover, I encouraged them to help students in discussions and debates in a controlled environment and give students the freedom to explore their learning. The results of my team were very encouraging.

For the Object-Oriented Analysis and Design course, we got students, who had studied almost three programming courses. However, these students were uncomfortable developing their logic and were hesitant to put their energies and efficient time into programming. Our methodology inbuilt the interest of programming in the students and helped them to overcome their deficiencies along with learning problem-solving and required concepts. For the Introduction to programming course, I suggested starting teaching using the Scratch software and later, programming concepts were mapped to activities in Scratch. I also introduced the idea of two mini projects, one from Scratch and one at the end of the course. I also initiated the use of simple animations as a programming example. A drastic increase in students' learning was observed.

Reflections on the pedagogical activity

When I started teaching, I followed a traditional approach, where a teacher teaches like a dictator and students act passively and merely take notes. In such an environment, students are not challenged to exploit their potential and feel no thrill of learning. They learn as a boring routine matter. The overall classroom is quite dull, and it is hard to see the spark in the students or the happiness of learning on their faces. However, this duration helped me to realize that there is something wrong. The feedback from students did not help to improve the course. I also realized that actual learning was more important than the contents delivered during a lecture. These factors forced me to look at alternative strategies and change my teaching style. Over the years, I have evolved a methodology that fills the problem/project-based learning and changed my classroom management strategies. Initially, I preferred to have more controlled classrooms. However, I now prefer classrooms, where people are more comfortable learning. I prefer more active students in the classroom than passive learners who mimic the concepts without learning them.

I categorize the subjects into two categories, programming courses and non-programming courses. It is worth noting that non-programming courses may also contain programming tasks; however, their purpose is not to teach programming, thus they are referred to as non-programming courses.

I consider programming courses as the foundation layer. I prefer a good blend of theory and practical work for programming courses. However, practical work is a more dominant part of the course. I follow two strategies for such courses, i.e., convert the less into more and teach to learn. For the first strategy, I teach the basic foundations of a topic and provide all the necessary information. Then, students must study the topic within a given time frame in given directions to understand its depth. For example, in a basic programming course using C++, I will explain to them that there is only one basic loop, i.e., a while loop.

The while loop may be used as a sentinel and non-sentinel loop (counter-based). Different examples are given from real life to programming. Later, for loop is introduced as an alternative to a counter-based while loop. At this point, students are required to explore the for loop and apply it to all the problems, which were previously solved with the help of a while loop. Students must also read about for loop and its usage from the book or Internet tutorials. Later, students are required to exchange their notes and to teach each other, which brings the second strategy, i.e., teach to learn. A group of students sit together, discuss the loop and teach each other what they learned.

In this way, what they learn becomes a threshold concept for them and changes their view. I believe in adopting different strategies to reinforce the learned concepts. A common strategy is creating an interesting problem set, given after a particular concept. Students are allowed to brainstorm in small groups to find solutions, however, they are required to devise solutions to enforce personalized learning. For a second, third or more advanced programming course, I prefer to follow a specific theme throughout the course. The theme helps in choosing the main example for the whole course.

Moreover, a series of small projects are given to students, which turn into a larger project later. This helps the student to realize the benefits of the divide and conquer strategy. Students are encouraged to work in groups of 2-4 or even a larger group (depending upon the available resources for the course) and come up with more than one solution to a problem. Students are also encouraged to learn soft skills in programming such as maintaining aesthetics while coding, reducing the number of lines and writing an optimized reduced code.

For the non-programming courses, a list of topics related to learning objectives will be prepared. Later, a main problem is found, which is mapped to topics. The theory is explained to students, and part of the problem, which is a complete problem, is given to students to solve. Afterwards, strategies such as in-class flipped classroom activities, and student-led presentations and discussions are used to reach the solution. Once the solution is found, students are encouraged to find the merits and demerits of their and others' solutions. Later, a small problem, which is not related to the selected theme or the problem at hand is given to solve. However, to solve this small problem, students need to follow the same steps as the solutions they discovered. This reinforces their knowledge and helps them explore the application of learned knowledge in another direction.

In both courses, a course book is used initially to develop the basics. However, later students are also exposed to recent research or the latest trends within the area. Depending on the course, students are given a series of mini-projects or a large project with multiple components. The projects are done in groups of 4-6 students (depending on available resources). For the project evaluation of both types of courses, students are required to demonstrate the application of learned knowledge for solving the problem and reflect on how to achieve the solution and the challenges they face. During the courses, an extensive availability time is announced for students to come and discuss their learning problems. Moreover, I prefer to keep students motivated through enthusiasm and excitement.