Teaching Philosophy

I do my best in providing the students the knowledge they need in the real world or showing the means to acquire it. As I mainly teach technical topics (like Embedded systems, Android programming) I have to make sure that students are motivated and attracted by learning such hard or low level topics. Therefore most of the time I do my best in finding real world examples and analogies to make students understand even the most abstract concepts. I also make extensive use of the blackboard for illustrating various topics and exemplifying/calculating as necessary. In a modern age, surrounded by powerpoints and passive teaching, I try to go back to the basics and teach in an interactive mode using the blackboard, asking questions all the time, giving tons of real world examples and making the classes and topics as interesting as possible. I also make sure that the important topics are well understood by students and I make sure to go again and again to key concepts if I think they are highly relevant. I also align my teaching such that what I teach during classes, what I expect them to know at the end of the course and what I evaluate during the exams at the end are following the same direction. I address active learning concepts (like problem based learning or question based learning) in my class by involving students in the teaching process as much as possible. Sometimes I use humor to captivate their attention and give funny examples related to technological issues and this sets a good mood in the class, which supports learning. I give plenty of opportunities for questions and I try as much as I can to be firm and secure in my answers. I promote a honest learning experience where I acknowledge my mistakes and accept corrections from students if this is the case. I know students are curious and therefore I try to find examples from everyday life modern technology, and give glimpses of how they work. In fact, this is why they chose such a technical profile, to learn how things work.
I try to use all the modern technology available (blackboard, teamviewer, interactive boards, powerpoints, on-line multiple choice questionnaires for training) and make sure students get rich content and explicit lecture notes, such that with little effort, the student can get the most out of it. Even though the topics are sometimes rigid, I promote the use of imagination in projects, in ideas and discussions. I think a balance between creativity and analytical work should bring the best out of the learning process even if we are talking about engineering. I also do my best in providing individual feedback to students (for all their work during the course) because I think it is important to remove the imaginary walls between the teacher and the student.
In the end, I try to improve all the time, both in the way I teach and in the contents I teach. I ask students for feedback regarding teaching and try to adapt to their needs and expectations. Some things need a few iterations to improve, others need a lot more effort. I am willing to grow and I know there is always a possibility for personal and professional growth. Even though I am the teacher in class, I never stop learning. This is also my biggest motivation for being in the academia.

Highlights of my teaching method

Question based learning

Giving students the opportunity to test their knowledge is in my opinion a strong learning facilitator. In the basic programming topics I usually like to address very simple programs and algorithms and I ask students to give me the answer: what is the outcome of this program from a list of choices, anonymously, by using an on-line anonymous interactive poll. I get their answers in one or two minutes and they can see the statistics in real time. Then, I do not give them the correct answer, but I ask a few students to briefly motivate why they chose a specific answer, so I get a possible explanation for each option. Then, I retake the vote and quite interestingly, the majority will get the real answer as usually it had the most plausible explanation. Such methods engage students and make them feel confident in their efforts.

Think before you test

I also like to teach programming quite old school, with pen and paper, before going to the computer and trying out things randomly until they work. I ask them to think like a computer would and in this way they get to analyze and understand much better programming concepts. It's hard at first, but after a while, it gets easier and more logical.

Individual assessment

I think is very important to be able to assess the performance of each student individually. For this, in the basic programming courses, I have worked quite a lot in having personalized theory tests (by developing a software tool that can generate such tests randomly). I use such tools to make sure I don't have any students that are able to "fly under the radar". I don't prohibit group work and collaboration during the semester, but I expect students to be able to handle programming concepts on their own. As programming requires not only knowledge, but also skills and competences for solving practical problems, I have a secondary evaluation method, in the lab, in front of a computer, focusing on individual work. I give them a problem they have not seen before and they are allowed to use their lecture notes, assignments as well as the internet to solve it, but not able to discuss or share their solution. I usually split the problem in 5 or 6 tasks, incrementally challenging such that the first tasks are very basic and set the threshold for passing my course. The end tasks require a little bit of skill and effort to solve, but they are not impossible. I usually have a bonus task at the end, which requires extensive knowledge and skills to solve, but it gives them extra credit. I sometimes corroborate both theory and practical problem solving results (where the students that got extra points in the practical part can lift their theory results - and it seems it is always the case) and I get a very good assessment of student performance.
For more advanced courses, I give students the opportunity to work on individual projects, which seems challenging at first, but they also learn the skills for self managing the workload and tasks. They also learn how to document things individually, which is a skill that requires quite some practice. Individual projects are very motivational, as I allow the students to steer the direction of the project (within reasonable limits) and they make great use of their creativity.

What should the students learn after my course?

They should learn both knowledge as well as how to apply the knowledge. To put it more visually: I think that knowledge represents dots and skills represent the ability to connect these dots. I want the students to be able to do both after following my courses, as this will set them on a very good start for professional challenges later on. I also try to prepare them for real world problems and very applied things, in today's economical context where engineers have to be flexible and be able to adapt by never stopping to learn.

Personal development over time

I started out my teaching activities as being insecure about the ways to teach and the topics I had to teach and I was going for a very conservative style. After seeing students' disinterest and boredom, I could easily see that I had to change something. Otherwise I would bore myself on top of not getting any interest from the students. This is something I try to avoid, as I know students can dislike topics just because they dislike the teacher. I started learning how to improve the teaching and after reading some pedagogical articles I have realized that how you teach is as important as what you teach.
I addressed different issues incrementally, and started to involve problem-based learning, question based learning and humor within the classes. I made the lectures very interactive and I can say that students like my classes.

How do I evaluate how the students see me?

First way: Students are very engaged in my classes and interact quite a lot. We have many dialogues, in class and outside class, formal and informal and I am always very open to anything they bring up.
Second way: I have a very good attendance rate to my classes
Third way: I always look to receive feedback on how I teach and student evaluations (both formal and informal) for the courses I taught have been very good.

Reflection

It is very hard to teach basic programming courses, as they require a lot of patience, understanding and a very interactive teaching mode from the teacher. I feel that is very important to have constant dialogue with students and to engage them in many practical and interesting problems. This makes them more motivated to learn.

Courses Taught

EMBEDDED SYSTEMS DESIGN I

Description:
This course offers students an introduction to the embedded world, addressing the basics of C programming language as well as providing a solid understanding on how microcontrollers work and how to deal with simple problems that can be solved with the use of a microcontroller.
Weight:
5 ECTS / 48h confrontation split 50% frontal teaching and 50% lab exercises
Target:
1st and 3rd semester bachelor level
Role:
Frontal teaching (lectures) as well as addressing the teaching during the lab sessions
Content:
I started from an older course taught at the University, but I had to improve the contents to meet nowadays-technological advancements and to address new and current real world problems and examples. Overall, a very practical oriented course with focus on providing the students with a strong foundation for any other programming courses that might be taught in their future studies.
Number of students:
Depending on the profile (Mechatronics, Electronics or Innovation and Business/Interaction Design), from 20 - 90 students.

EMBEDDED SYSTEMS DESIGN II

Description:
This course offers students a deeper look at the embedded world, addressing some of the advanced functions of microcontrollers (timers, peripherals, interrupts) and how to deal with more complex problems that can be solved with the use of a microcontroller. It also gives a more solid theoretical background for hardware design (logic gates, state machines).
Weight:
5 ECTS / 48h confrontation split 50% frontal teaching and 50% lab exercises
Target:
2nd and 4th semester bachelor level
Role:
Frontal teaching (lectures), as well as practical teaching during the lab sessions. If the number of students is high, sometimes I get help from a teaching assistant for the lab sessions.
Content:
I started from an older course taught at the University, but I had to improve the contents to meet nowadays-technological advancements and to address new and current real world problems and examples. Overall, a very practical oriented course with focus on providing the students with a strong foundation for advanced hardware problems.

Number of students:
Depending on the profile (Mechatronics, Electronics or Innovation and Business/Interaction Design), from 20 - 90 students.

Evaluation (both EMBI and EMBII):
Depending on the profile and how their grades are weighted as part of their curricula, as a combination of written (theoretical) with multiple choice + short answers or simple schematic solutions + practical problem solving in the lab, time limited, but open book/internet access. If an oral exam is part of the curricula, these results are filtered through an oral evaluation where students get the opportunity to defend their work on both a practical and theoretical level.

Improvements (both EMBI and EMBII):
I have always looked on improving the course on hardware level (boards to be used - from old Phillips microcontrollers to Atmel), providing a cheaper and technically superior solution that makes the practical part more appealing to students as well as saved the University licenses and registration costs. The platform change had a high impact for the whole study, as students used the same platform in other courses. This platform can be seen as a step forward towards current day technology and allows the students to develop extended skills and to address new kinds of problems/practical applications. Recent surveys for embedded engineers show Atmel as one of the key market competitors for 8bit microcontrollers and provide an excellent starting platform for students.

DEVELOPING APPLICATION FOR MOBILE PLATFORMS

Description:
This course offers students an introduction to Android for mobile platforms, addressing the basics of Java programming language as well as providing a diverse palette of practical applications related to engineering (controlling actuators, reading sensors, user interaction etc.).
Weight:
5 ECTS / 48h confrontation split 50% frontal teaching and 50% lab exercises
Target:
4th, 6th, 8th semester students (Mechatronics, Innovation and Business/Interaction Design)
Role:
Frontal teaching (lectures) as well as addressing the teaching during the lab sessions
Content:
A fresh course in a very interesting technology and following state of the art tools and concepts. It is continuously upgrading according to the advancements in mobile technologies, but strongly related to how mobile platforms (tablets, phones) can be used in combination to other engineering projects within their study.
Number of students:
Depending on the profile (Mechatronics or Innovation and Business/Interaction Design), from 20 - 55 students.
Evaluation:
Oral exam based on defending individual projects. The individual projects are chosen by students themselves and allow them to try to solve/tackle a real world problem or implement an idea they have. Self-driven projects are a very motivating way to learn a new platform.

EMBEDDED SYSTEMS DESIGN - ESD (master level)

Description: The course addresses software design and development as a science, UML design artifacts, SCRUM, use cases, software architectures as well as modeling of hierarchical state machines and their deployment onto hardware platforms.
Weight:
5 ECTS / 48h confrontation split 100% frontal teaching and in class exercises
Target:
8th semester students (Mechatronics, Innovation and Business)
Role:
Frontal teaching (lectures) as well as addressing the teaching during the exercises
Teaching aspects:
The course is very theoretical and is based on many abstract concepts, which are sometimes very hard to follow, especially by students looking to write code. I usually try to show and discuss a lot of alternative and real-life examples and engage students in trying to find similar examples of their own. It is also very hard to teach designing state machines only with PowerPoint presentations, so I take a very practical approach where I present them the development of a basic state machine from scratch, explaining all the steps as I go along. I also use dialog for interaction and asking students to contribute with their thoughts of how should the problem go. It is more like an interactive tutorial. Given the relative small size of the class I can get by using the projector
Number of students:
Between 15 - 20 students.
Evaluation:
Oral exam based on picking a selected topic as well as questions about a state machine.

SEMESTER PROJECT: BUILD MECHATRONIC PRODUCTS THAT CAN MOVE

Description:
The semester project offers students to possibility to integrate the knowledge acquired on basic courses to build a mechatronic device that can execute a specific task (different every year). The end result is some sort of robot that is able to move by using sensors, actuators, smart body design and different control techniques. As the intelligence part as well as sensor reading and motor control are related to the EMBEDDED DESIGN course that I also teach, I try to correlate as much as possible the contents, knowledge and abilities of the students from both these courses.
Weight:
10 ECTS / weekly lectures or group meetings
Target:
2nd semester students (Mechatronics)
Role:
Frontal teaching (2-3 lectures during semester) as well as coordinating group meetings every week + help during project period. The workload is split in 2: I mainly supervise and assist in the embedded part, while a secondary teacher supervises and assists in the mechanical part.
Content:
Groups of 3 students are addressing the same problem, but with a different project. The project combines concept design, proof of concept, implementation, mechanical design, embedded design as well as integration, group work challenges and writing engineering reports.
Number of students:
40 - 50
Evaluation:
Oral exam based on defending individual projects.

BACHELOR SUPERVISION

According to student wishes and my research interests, I try to find a balanced topic that uses their skills at this stage and is very motivational for them as well as well as addressing some very practical challenge of some sort. I usually set bi-weekly one-hour meetings to address the progress and future work with each student. I have tried to implement basic artificial intelligence in miniaturized electronics for real time applications and the results have been very good, both student satisfaction as well as learning outcome.

PROFESSIONAL BACHELOR SUPERVISION

Following topics set by industrial partners, I supervised students for their 7th semester when having an internship in the company. These projects are very practical and focus on solving various issues that are of interest to the companies in the area. I offer students guidance for solving these tasks and help them understand the engineering method (find the problem, find solutions and alternatives, test them and pick the best) as well as guide them in documenting and planning their work. Meetings are according to their company schedule, but usually 2 hours every 2 weeks with each student. Topics addressed so far are within production optimization in assembly lines or industrial automation.

MASTER SUPERVISION

According to student wishes and my research interests, I try to find a challenging topic for the students to explore. I give them more freedom and expect them to work more research oriented as well as work independently. I still address weekly one-hour meetings every week, to keep track of the work and results so far. I also help them with writing excellent documentation and the tools for that (latex). The topics addressed so far included: applied artificial intelligence (detection and classification of surfaces using tactile sensors, real-time detector of contaminated food bags for automated packaging, online recognition of engagement of people), smart sensor development, advanced and miniaturized electronics as well as autonomous navigation, exploration for robots and flying drones.

Pedagogical qualifications

Followed the Lecturer Training Program (LTP) given by University of Southern Denmark in 2013 - 2014 where I learned a lot about how to improve my teaching.