Theoretical Computer Science for Diverse-Skilled Students

Yehia Abd Alrahman

May 25, 2023

Theoretical Computer Science for Diverse-Skilled Students

Yehia Abd Alrahman

Department of Computer Science and Engineering
Chalmers | University of Gothenburg, Gothenburg, Sweden
yehia.abd.alrahman@gu.se

Abstract. Theoretical Computer Science (TCS) classrooms are known to be very challenging to handle. The difficulty is not directly related to the technical nature of the courses, but rather to the diverse skills and knowledge of attending students. These courses are offered to computer scientists, engineers, or even students with major in mathematics. The diversity of students makes it hard to assess how to efficiently engage students by introducing appropriate teaching activities in the classroom. Some of the major difficulties are connected to inclusion, group formation, and type of assessment. This paper aims at exploring the teaching techniques used in the field and identify strategies that best fit and align with TCS classrooms. In this paper, we will reflect on a TCS course, named "Testing, Debugging, and Verification (TDA567/DIT084)", that the author is currently responsible for, and teaches for a mixture of students from Chalmers University of Technology and Gothenburg University.

1 Introduction

Teaching theoretical Computer Science (TCS) courses is not an easy task [5,47]. The main difficulty stems from the fact that the target audience of such courses is inherently heterogenous. Indeed, students, attending TCS courses, have usually majors in either computer science, engineering, or mathematics. Despite their heterogeneity, their profiles are still complementary and essential to deal with the contents of such courses. This heterogeneity, however, implies that the progress of such diverse groups will vary during the course. The goal is to find teaching methods and strategies to ensure that all students are actively engaged throughout the different stages of the course. Moreover, there is a special interest to trigger agency in students to allow them to build on their prior experience [51] to reconstruct a deep understanding or a personalised reformulation [12] of the problems they encounter.

Traditionally, TCS courses are approached in a form of "Sage on the Stage" practice [18] where the teacher (or the lecturer) is the center of the process and the students are passive receivers. These traditional approaches follow a transmittal model of learning [27]. That is, the model assumes that student's profiles are homogeneous and their brains are nothing, but empty (or identical) containers. The teacher has the ability to pour knowledge "evenly" in such containers. Clearly, this view is in contradictions with the constructivism model [51,9], a prevailing model in modern learning theories. The latter recognises the knowledge gap among the different students, and casts learning as a process of knowledge construction based on an active reflection of current and past experiences of the students [35].

Note that traditional teaching methods are not to be considered harmful or "bad", however, they cannot be applied effectively in the diverse TCS classrooms. This is because that these practices treat all students as if they have nearly identical skills, motivation, and prior knowledge about various subjects. As a consequence, they negatively impact on the interaction with the teacher and the ability to think critically and construct new knowledge. Clearly, TCS teachers must be equipped with "teaching toolkits", grounded by research-validated strategies that motivate and engage every student in the learning process. Moreover, these toolkits must provide strategies to foster collaboration among diverse student profiles.

In this paper, we focus on active learning strategies [41] that respond to the variance in interest, knowledge, and education profile [46] of TCS students, and show a possible implementation. The latter focus is due to the fact that active learning is well-recognised in our field (and more generally in natural sciences). Moreover, various research studies show that active learning can possibly increase engagement of students [43]; can increase retention and deep understanding [4]; can improve and promote both performance and problem-solving in students [17]; and allow students to stay-aligned with the learning objectives [32]. However, active learning does not necessarily respond to the diversity challenge in TCS students. Thus, we limit our attentions to active learning techniques while assuming that students, attending TCS courses, have varying amounts of prior knowledge about any given topic in the course or their background in mathematics.

We are not looking for developing basic cognitive skills like memorising or applying the acquired information (cf. Bloom's taxonomy of cognitive skills [3]), but rather we look for strategies that enable the students to challenge existing techniques or formulate personalised versions of the problems they learn [12]. We also want to exploit the diverse-profile collaboration to serve the purpose of learning.

Throughout this paper, we will reflect on a TCS course, named "Testing, Debugging, and Verification (TDA567/DIT084)", that the author is currently responsible for, and teaches for a mixture of students from Chalmers University of Technology and Gothenburg University.

The structure of this paper proceeds as follows: in Sect. 2, we explain the diversity challenge by reflecting on the course "Testing, Debugging, and Verification". We also explain our expectations from a viable solution; in Sect. 3, we present active learning in relation to diversity, and the main active learning strategies that we focus on in this paper; in Sect. 4, we propose a course design that responds to the diversity challenge and use it for the course "Testing, Debugging, and Verification"; in Sect. 5, we present concluding remarks.

2 The Diversity Challenge in TCS Classrooms

In this section, we discuss the diversity challenge in TCS classrooms in more details. We clarify the context in which we consider diversity, discuss the main problems, and acknowledge the importance of differentiated instruction as a possible mitigation strategy. We also present our view of a possible cost-effective implementation of differentiated instruction in terms of time and resources. To capture the reader's intuition, we will use a TCS course as a running example.

2.1 Context

We consider the diversity issue of TCS students in terms of variance in learning profile. That is, the different levels of knowledge, skills, and interests in various subjects (cf. [10,44]) due to different education paths. This is different from the issues related to gifted education or students with learning difficulties, as heavily investigated in the literature (cf. [20,52,34,37]). The former problem makes it hard for teachers to design effective teaching activities, ensuring that every student has an equal access to the course contents. Thus, we may need to shift the focus from the traditional definition of constructive alignment [6], that requires aligning the learning activities and assessments with the learning objectives, to actually align (or differentiate [46]) the learning activities with the learning profiles of attending students.

2.2 Showcasing

To exemplify the problem, we will reflect on a lightweight[1] TCS course, named "Testing, Debugging, and Verification (TDA567/DIT084)" (or TDV for short).

The TDV course [2] is composed of four modules, namely Testing, Debugging, Specification, and Verification. The first two modules feature lightweight math contents while the last two modules mainly contains mathematical descriptions and proofs. The course aims at providing basic understanding of the techniques that cope with errors in computer programs. The course is 7.5 academic credits, divided into 5 credits for teaching the underly theories of the modules (mostly mathematics); and 2.5 credits for laboratory work which consists of projects for each module. The course syllabus indicates that the course can be part of a Computer Science Program (CS), Applied Data Science Master Program (DS) and Software Engineering and Management Master Program (SE). Entry requirements specify that students must have successfully completed a course in discrete mathematics and a course in programming.

The structure of the course is divided into lecture sessions, weekly exercise sessions, and weekly lab sessions. The exercise sessions consist of small exercises based on lectures in the same week and can be solved in groups. There are four sizeable group laboratory projects, one for each module. To pass the course, students have to pass both laboratory projects and a final written examination about the theoretical concepts.

The population of potential students is diverse and serves a mixture of skills and interests. For instance, computer scientists usually have better algorithmic skills while software engineers have better design skills that are essential for architecting large software. Moreover, data scientists are more mathematically inclined with programming skills tailored to scientific tools and libraries.

Due to the diverse profile and education paths of attending students, the evaluation of the TDV course usually gives the impression that this course divides the students into three different groups; either enjoying the whole course (mostly computer scientists), only the lectures (mostly data scientists), or only the lab work (mostly software engineers). The former labels are based on an informed guess of whether the students have interests in theory, practice, or both.

For instance, in a recent iteration of the course, the students responded to the question "What should be kept for the next round of this course?" as follows:

> "All of it!"

[1] The course serves as an introduction, and thus does not dive too much in technicalities.

> "the labs"

> "the in person lectures", "the exercises where great"

In response to the question "What is your overall impression of the course?", samples of students answer were as follows:

> "This is one of the few courses where I truly feel that what I am learning and spending time on is useful"

> "Not very challenging but still some learning outcomes. Doesn't push any boundaries though"

> "Weird focus, not enough testing, outdated debugging and a lot of focus on formal verification"

Clearly, the answers highlight the differences by subject background. CS students find interests in both theory and practice while DS students look for mathematical depth which was not met by this introductory course. On the other hand, SE students are more practical, and thus wanted the focus to be more on programming related practices.

2.3 Main Challenges

Experience in teaching this kind of courses indicates two main intertwined challenges that teachers find hard to tackle. Namely, group formation and efficient assessment. The interplay between these two can negatively impact on student's inclusion. Below, we discuss each one separately.

Group formation. Efficient Group formation for diverse-skilled students is not an easy task. Current practices vary depending on the student skills[2], their preferences, and their personal constraints. For instance in several iterations of the TDV course, the groups were formed as follows: some students choose their groups based on mutual interests; or based on personal constraints; and others make decisions randomly on the spot. The problem here is that such groups suffer from stability issues. Indeed, some groups are dissolved in a later stage either because of disagreement or because specific students want to work individually due to personal situations. We have students with special needs, part-time workers, etc. The teacher tries to accommodate their situations rather than being exclusive. Other forms of disagreements is usually because students with similar interests re-group and locally divide the group into different coalitions, isolating less capable students. Moreover, some students from non-europeans cultures do the entire lab work without any help from their group members because they do not think it is appropriate to speak out.

As mentioned before, the lab work is composed of four sizeable projects, one for each module. The first two projects are more accessible to students with programming backgrounds while the last two are more accessible to students with mathematical backgrounds. Due to unstructured group formation, students of different backgrounds felt disconnected on different stages of the lab work, and suffered from difficulties delivering their projects. Clearly, there is a need for a better strategy for group formation ensuring group -stability, -functionality, -delivery, and -individual accountability.

Assessment. Assessing diverse-skilled students is also very challenging. The main goal of assessment is to make sure that the progress of the student is well-aligned with the learning objectives of the course. However, assessments are not exhaustive and should not be. Thus, we need an accurate and unbiased way to assess students while addressing the variance in their learning profile. The latter proved to be hard as computer science and mathematics share concepts and notations that might be interpreted differently, and thus make assessment a complicated task.

For instance in the last iteration of the TDV course, there was a divide among the students on the quality of the final written examination. Here are some of the student comments:

> "the exam was good and too easy to check if I actually understood the topics"

> "the if-test is confusing: if $(x[i]\%2 == -1 \mid x[i] > 0)$, but any integer either positive or negative modulo 2 if odd return 1, so I do not understand what you mean"

> "luckily I do know my math and modulo but unluckily I did not know enough about the % operator in Java!"

[2] Currently, teachers of the TDV course do not have a clear idea about the profiles of attending students.

The last two comments are about the % operator in the Java programming language which is used in the TDV course. For Java programmers, it is obvious that this operator computes the reminder of division, e.g., $(-1\%2) = -1$. The latter is a correct answer in Java, but does not make sense in mathematics if the % operator is interpreted as a modulo. That is, under a mathematical interpretation, we have that $(x\%2) = 1$ for all odd numbers (positive or negative).

It turned out that both comments came from students with major in Data Science who mainly use Python as a programming language. The latter interprets % in the mathematical sense. Note that this course is not about teaching programming and it assumes that students already completed at least 7.5 credits of programming. Thus, it is justified that the teacher did not foresee the confusion.

2.4 Mitigation Strategies

To tackle the above mentioned challenges, we need to identify appropriate mitigation strategies that can be used to lay the basis for a cost-effective teaching model of TCS classrooms both in terms of time and resources.

Differentiated instruction [46,45] has been proposed to tackle diverse classrooms in general. The focus of differentiated instruction is on varying the instruction regime (e.g., course contents, instruction methods, assessment, etc.) in response to students readiness, interest, and learning profile. The rationale is that students learn most effectively in their zone of proximal development [53], i.e., what students are able to do with a bit of guidance. The literature on how beneficial it is to actively differentiate instruction in response to students readiness, interest, and learning profile are overwhelming (cf. [46,39,28,7,29]). However, a direct implementation of differentiation is not usually attainable due to limitations of resources, i.e., few teaching staff and timing constraints.

Our view is to use well-established teaching methods that feature collaboration and peer teaching to cope with resource limitations. This is inline with the view of the main pioneer in differentiated instruction [46,45], Carol Ann Tomlinson. She suggests that an effective differentiation of instruction should: (1) be proactive rather than reactive, namely the design of the teaching activities must be done in light of student's variance; (2) exploit workgroups in the classrooms; (3) avoid "one-size-fits-all" strategies, namely by varying pacing and teaching materials for both groups and individuals. This is important because students learn most effectively in their zone of proximal development [53]; (4) be knowledge centred, namely to focus on essential knowledge and skills that can be used to solve real and meaningful problems (i.e., do not obsess about coverage); and (5) be learner centred, namely by focusing on the student's needs, prior knowledge, motivation, engagement (i.e., by allowing the student to make sense of what they learn).

In our view, active learning techniques that feature collaboration and self-maintaining group work can be an excellent candidate. This is because they bypass resource limitations by distributing the learning process among self-directed groups, including the teaching team. Thus, we will focus in this paper on active learning as a tool to differentiate instruction according to Tomlinson's characterisation.

3 Active Learning as a Tool to Differentiate Instruction

We focus on active learning strategies that the teacher can use to differentiate instruction for diverse-skilled students, while optimising the use of resources. Our understanding is that nurturing student groups that function autonomously is the key to cope with limited -time and -staff for the teaching sessions. Thus, we will review curricular design approaches that are based on collaborative active learning ideas. We start with small-scale technique (like Menti Quizzes [24] in lectures), and then we lift the focus up to thinking about the bigger picture of a complete course design.

Active learning. Active learning [41] is an interactive teaching method that follows the concept of "*learning-by-doing*", through a set of meaningful and engaging activities in which the student get encouraged to actively participate and develop a sense of ownership to learning [12]. Students are given the lead on the learning process, and get encouraged to become the actual protagonists of their learning. Teachers play a mentorship role and ensure that the activities of the students are well-aligned with the learning objectives. Moreover, teachers are responsible for facilitating and evaluating the learning process.

Various research studies show that active learning can possibly increase engagement of students [43]; can increase retention and deep understanding [4]; can improve and promote both performance and problem-solving in students [17]; and allow students to stay-aligned with the learning objectives [32]. However, active learning methods are not an act of individuals but rather the result of a collaboration of various entities such as well-planned activities from the teachers side and motivation and engagement from the students side.

Technical challenges. There are still various obstacles when it comes to an actual implementation of an active learning model. As mentioned in [8], the main challenges that educators face when implementing active learning in classrooms are: the scalability problem, i.e., traditional active learning methods do not scale well as the classroom size increases. That is, engaging a large number of students in a limited time can be hard; the "coverage" problem, i.e., the tradeoff between the amount of active learning in classroom versus covering the learning objectives; the timing problem, i.e., increased class preparation; the heterogeneity problem, i.e., the variance in knowledge, interest, and motivation in students; the resource problem, i.e., limited, or lack of resources and support; and finally the risk problem, i.e., disapproval from colleagues, student dissatisfaction, and the impact on promotion and tenure decisions (in case of bad evaluation).

These challenges have been investigated heavily in the literature (cf. [17,36,5,23]) and many solutions have been proposed. However, most of the proposed solutions either ignore the heterogeneity problem of students (cf. [36,5,23]) and target active learning in general or consider heterogeneity as another problem that can be handled separately (cf. [50,42,14]). In practice, these two problems are intertwined and their negative impact increases by a magnitude of factors in diverse profile classrooms such as theoretical computer science. Indeed, the interplay between these two problems results in new problems related to inclusion, group formation, type of assessment, etc. For instance, inappropriate design and planning of active learning activities results in excluding students with insufficient background, random group formation can lead to a situation where the group is locally divided due to regrouping based on interests, and lastly, the interpretation of the assigned tasks might lead to confusion due to the different backgrounds of students. Thus, there is a pressing demand to study the different approaches and find strategies that best fit and align with TCS classrooms. Below, we discuss the relevance of the different active learning curricular design approaches to TCS classrooms.

3.1 Small-scale approaches

Small-scale approaches of active learning usually take the form of sporadic interventions during lectures. Some of the tools that are popular in computer science are Menti Quizzes [24], Hot Seat [13], Shortlists and Error Hunt [19], Think-Pair-Share [26] etc.

These approaches use active learning as a supplement to traditional lecturing rather than going to the extreme of a complete course re-design [21]. When experimenting with active learning for the first time, these methods are recommended as they are safe to use. That is, these methods are lightweight, and thus the teacher has time to react if things go wrong. The latter suggests that such teaching interventions are reactive, and assume that problem resolution is to be done during the lecture. This is, however, can be tricky in diverse-skilled classrooms as time is limited and the space of errors is large. Indeed, the design of teaching activities for diverse-skilled classrooms must be rather proactive, as Tomlinson [46,45] suggests. That is, the activities have to be designed in light of the student's variance.

Moreover, the structure of the class and the setup of these activities is rather spontaneous. For instance, when using Think-Pair-Share during class-time, we cannot make use of mixed-skills collaboration as students with similar interests tend to sit closely. It is also not clear how to exploit these methods to aid efficient assessment of diverse-skill students or to encourage peer-teaching in a mixed-profile settings.

Indeed, in the most recent iteration of the TDV course on Winter 2022, the author became responsible for the course and started the process of transforming the lectures into active learning exercises. The author started with sporadic active learning tools such as Menti Quizzes [24], Hot Seat [13], and Error Hunt [19]. This was in a response for an evaluation from a previous iteration where some students expressed that they had difficulties understanding some algorithms in the course because the lectures were less interactive. However, the last evaluation of the course featured similar divide among students, and thus it did not provide an idea if students understanding was improved.

We became more convinced that a complete re-design of the TDV course that supports proactive intervention strategies is on demand.

3.2 Large-scale approaches

As the name suggests, large-scale approaches of active learning consist of a complete re-design of the course and the teaching activities. The main idea in these methods is that class-time is not the only resource for learning. Blended Learning [22] strategies can be adopted to make use of student's time before the class. In other words, class-time only serves as an opportunity to assess and mentor students progress rather than to teach them new things. Most of the work is done before the class through well-structured and self-directed student groups that function autonomously. Cooperative-based learning [4] is one of the most used large-scale active learning in natural science. The advantage of such methods is that they can mitigate the challenge due to shortage of resources. Moreover, the group work provide an opportunity to make use of the diverse profile of students to encourage peer-teaching and nurture collaboration in a multidisciplinary settings. However, the design of the course must be studied and analysed thoroughly to avoid large-scale failures that are hard to account for during the course.

Below, we will discuss how to use these approaches to support diverse-skilled classrooms and address the challenges of efficient group formation and assessment.

Cooperative-based learning Cooperative-based learning is built based on the fact that learning is a social activity where interacting and learning from peers have a positive impact on the cognitive development of the student [4]. Clearly, group work is not only about load division, but rather the fact that a well-cooperative group can do more than that of an individual. The latter creates a sort of promotive interaction where students help and facilitate one another's work in the group. Moreover, management and leadership skills can be naturally nurtured. Indeed, students learn how to better communicate, facilitate one another's work and handle conflicts. Students also learn to think as a team and reflect on the functioning of their team, and look for ways to improve and optimise their learning process.

Cooperative learning is built to support "individual accountability" and "peer-assessment" in the group. The latter are recurrent problems when teaching the TDV course because the groups are usually observed as a whole, and the average behaviour of the group shadows the one of the individuals. In cooperative learning, each member has a role in the group and the group as a whole perform peer evaluations and mentoring that can be useful for teachers to grade an individual in isolation of the group. This method is largely used in the bachelor thesis project at the department of computer science and engineering, Chalmers University of Technology [1]. Despite its success and extensive documentation, the author is not aware of any course that adopts the approach at the department.

Cooperative-based learning frameworks that provide better ways to construct group work and enable efficient assessment can be ideal for TCS classrooms. However, there are various elements that need attention. Clearly, cooperative learning is more than simply assigning students to groups. Indeed, students may not even understand how to work as a team until they actually practice it. For this reason, early encounters with the students are used to explain the method and teamwork principles. Moreover, the formation of the teams is discussed where the teacher explains the rationale behind it. For TCS classrooms, teams must be formed in a way that takes into account students' interests, abilities, previous knowledge. Teams must be stable and foster peer teaching among students. That is, its members have diverse profile that encourage positive interdependence rather than local regrouping and conflicting coalitions. Teams must be formed to function as an autonomous entity that self-manages its members in a decentralised fashion. Namely, the control must be distributed among team members to avoid a situation where one or two students take control and isolate the rest of the team.

Cooperative learning is the umbrella that covers many concrete methods such as *project-based learning* [30] , *problem-based learning* [15] (or *inquiry-based learning* [16]), and *team-based learning* [21]. Each method can be used to serve specific needs, as explained below.

Project-based learning. The essence of this method is that students can construct new knowledge on demand, while trying to solve a sizeable problem (or a project). Initially, students use their previously acquired knowledge to approach the problem and based on what they encounter they research new challenges and construct new knowledge as they go. The attractiveness of this approach is that it puts the learning objectives in context and foster problem-solving skills. That is, it gives meaning to why the students need to learn new concepts and how to apply and/or adapt them appropriately in different settings. It is worth mentioning that the choice of the project, the team, team dynamics, and type of assessment will majorly impact on the quality of this active learning method. Indeed, a poor planning from the teacher can render this as a cookbook where the students approach the project with a recipe in mind, and thus failing to accommodate critical thinking and problem-solving skills.

Projects are usually sizeable and target a coherent story, and thus if we use them in a cooperative learning framework, they can be ideal for a TCS course in which its contents can be explained through a large case study.

Problem-based learning. Problem-based learning [15] (or Inquiry-Based Learning [16]) is majorly used in medical sciences, and also in mathematical proof-based courses. As the name suggests, it is an approach based on questioning and investigating problems to reach the truth. It follows the inductive principle of problem-solving in that it starts with current knowledge and later seeks to construct new one. Unlike Project-based learning, it is usually used in small-scale problems that can be solved during the class.

A possible implementation of this approach is as follows: the teacher pose a small (theoretical) problem and the students discuss it in small groups based on their current knowledge. Later, the teacher asks questions related to the problem and individual students answer them. Group meetings are arranged after the class so that students discuss further and agree on a solution. The teacher mentors and facilitates group work and makes sure that the groups are functioning properly.

As mentioned before, the targeted problems in this approach are usually small in size, and thus we can use this approach to teach theoretical concepts and to lay out the conceptual framework of a TCS course. Indeed, many TCS courses that cover background materials (heterogeneous in nature) for more advanced courses can benefit from this method, e.g., Discrete math, Automata theory, Computability, etc. For instance, the constructions in a course on Automata theory can be used for an advanced course on Model-Checking. The reason here is that content-coverage in such courses is important, and due to heterogeneity of contents, it is hard to design appropriate projects.

Team-based learning. Team-based learning [21] is concerned with developing teamwork skills more than actual problem-solving. It is heavily packed with tests to hold students accountable for their performance and their contributions to the team. That is, students have access to the course material beforehand, and they take individual readiness test at the beginning of the class. Later, they take the same test with the group, and the grade is a combination of what they achieve individually and with the group. The rest of the class time is organised as a sequence of group exercises.

Although this approach sacrifices research- and inductive- problem solving, it actually solves a major problem regarding feedback and/or assessment. That is, this approach provides a precise feedback for the teacher to measure the contributions of individual students. Indeed, successful group test only informs on the average performance of the group as a whole, but is insensitive to individuals. Statistically speaking, average is biased to large numbers in that a highly performing student in a small group will shadow the low performing ones in the same group (see [11,49,33]). This implies that the feedback gets more accurate as the group's size gets smaller. Thus, initial individual test will give the teacher a clear understanding on how to both assess and give precise feedback to individual students.

In general, some ideas about assessment can be adapted from this approach to help evaluating a TCS course.

In the next section, we will build on ideas from cooperative-based learning techniques, and specialise them to a lightweight theoretical computer science course that features diverse-skilled students.

4 The TDV Course: A Differentiated Active Learning Design

We propose a differentiated active learning design for a lightweight TCS course that the author is responsible for. The design decisions build on a previous teaching experience of the course and the literature recommendations of how to approach the student's variance [46,45], and active learning practices in both engineering and mathematical disciplines [36,5,23]. We will focus mainly on three aspects, namely: class setup, learning activities, and assessments.

We propose a combination of project-based [30] and problem-based [15] learning in a cooperative settings [4] as the main teaching methods. We believe that a course design based on cooperative learning is beneficial when dealing with mixed-skilled students. That is, by forming mixed-skilled autonomous groups, we not only can bypass the resource limitation challenge, but also provide opportunities for peer-teaching and multidisciplinary collaborations. Moreover, we can adopt existing assessment strategies to evaluate students based on group work and individual contributions. In other words, a student is not required to majorly contribute to every part in the group project, but at least to the aspects that match their background.

We use project-based learning as a means to allow students to do groupwork on sizeable projects towards the end of the course. The use of project-based learning is not only motivated by its successful adoption in mathematical and engineering disciplines (cf. [38,48,25]), but rather because the structure and the contents of the course is a perfect match. The topics of the TDV course are the basic building blocks for software developers to gain confidence in their products and provide assurances for the target users. Thus, they form a coherent story that is approached in a stage-by-stage fashion.

As mentioned before, groupwork is more than simply assigning students to the groups. Students may have knowledge differences, and thus we need a way to ensure that students are ready to function in the groups. For this reason, we use problem-based learning as a way to lay the conceptual framework, engage prior understanding, and trigger agency and self-monitoring in students (Meta-cognition) [40].

4.1 Class Setup

Due to the variance in students profile and education path, we will design an attributed questionnaire and ask each student to fill it one week before the start of the course. The goal of the attributed questionnaire is to collect information from the students about their interests, skills, and education path. The latter will be used as a means to align the project contents and form the groups.

The first lecture is used as a roadmap of the course format, teaching methods, administration stuff and the examination. The teacher explains project-based and problem-based learning and their objectives in detail. The teacher makes the students aware that this course covers a diverse profile materials that build on each other to serve a useful purpose. Thus, it is natural that the students will need to depend on the diverse skills of their group members to accomplish specific tasks. It is also natural that different students will take lead on different aspects of the project based on their skills and competence. This will justify the use of the attributed questionnaire and why the teacher is the one who forms the groups and chooses the projects per groups.

Projects. Due to resource limitations, we will only use three different projects to be distributed to the students (one per group). This is to allow differentiation of teaching for different groups in response to variance in interests, skills, and education paths. Although, the groups must be diverse by nature, we can not always make sure it is the case. This is because that the diversity degree of attending students is not under our control. Thus, depending on such degree, we provide projects that best align with the distribution of students profile.

Group formation. We form the groups to be autonomous entities that are functional, self-managing, and self-monitoring. That is, we distribute the control by assigning individual roles for the different members. The latter can be implemented by using a *project contract*, namely the rules for cooperation and individual responsibilities that the students agree on. The contract must be prepared in writing and be present for the teacher. An example of such contract can be found in [1], annex 1. We also want to ensure that the groups are stable, foster peer teaching, and encourage positive interdependence. By using heterogeneous groups with aligned projects, we create an opportunity to realise such goals, but we cannot enforce them. The latter requires incorporating individual accountability in the group. For this reason, we propose to use *project diaries* (cf. [5,1]) and *contribution reports* [1] which are suitable for heterogeneous classes (See examples in [1], annex 2).

Project diaries are used to record the plan of the project in terms of individual responsibilities, milestones, time reporting for working hours of each member, and a chronological log of the different stages of the projects, including encountered problems, proposed solutions, and reflections.

Contribution reports, on the other hand, can be used to account for individual contributions of the students. Each student in the group ranks other group members in terms of their contributions for the different parts of the project on some scale, and the report is submitted on different stages of the project. For example, students can be given a figurative budget, and asked to distribute it as salaries to the different members according to their contributions. The teacher can use the latter as a way to provide individual grades. This approach provides an equal representation of every member of the group and helps ensure stability, i.e., no member has a higher control on the group.

Course and reading materials. all course materials (including lectures, assignments, and suggested readings) will be available online before lecture time. We will use short pre-recorded lectures (one per topic alongside with a group assignment), prepared in Blended Learning format [22]. We will differentiate reading materials to account for the variance in students profile. For instance, we provide readings with applications in mathematics, computer science, and software engineering. Moreover, assignments and reading materials will be chosen to narrow the research efforts that the students must go through to finish their projects. For instance, assignments will be designed to be aligned with the group projects. Thus, the students may reflect on the solutions of such assignments and build on them to solve larger problems they may encounter during their projects.

4.2 Learning Activities

We propose two types of learning activities: small group assignments to lay the conceptual framework and engage prior understanding; and sizeable group projects that nurture collaboration and problem-solving at a larger scale.

Students go through reading materials, pre-recorded lectures, and group assignments individually before class time. Later, students arrange individual group meetings to discuss and solve the assignments. The lecture is normally divided into time-limited slots, one per assignment. For each assignment, one group is responsible for presenting the assignment and the proposed solution within the specified time frame. Other groups must ask questions, provide feedback, alternative solutions through representatives, and the presenting group must defend their solution. Groups can alternate presenting the assignments based on a round-robin scheduling [31]. The teacher chairs the discussion, clarify ambiguities, and correct misconceptions.

Group projects, on the other hand, are sizeable, do not have unique solutions, and can be approached by different techniques. Thus, students need to justify their proposed solutions. The idea is that students will not only learn by doing, but also will own and defend their results. The main motivation of using open-ended projects is that different groups learn different things, share, and consolidate their learning during in-class discussions.

Group projects are divided into four stages, one per module in the course. Groups submit progress reports for each stage and receive continuous feedback from the teacher and the TAs until the end of the course. Students will always have the opportunity to ask the teaching team questions, and the team will answer in way that narrows the research parameters for them.

Towards the end of the course, groups sign up for 30 minute sessions each. They must present their project and their design choices. Students from other groups must critically discuss the proposed solution, highlight weaknesses, limitations, etc., and support their opinion by evidence. For example, in terms of coverage criteria for software testing, the opponents can provide "test cases" that the proposed solution does not cover. They must propose alternatives and argue for that. The presenting students must defend their choice and argue for it. After the final seminars, all groups submit their revised projects and final reports.

The groups consists of mixed-skilled students, and thus students are asked to take lead on group work that matches their skills. The latter would allow each member to understand their role and value in the group, and possibly creates peer-teaching and positive interdependence.

4.3 Assessment

Our experience from written examinations of the TDV course clearly indicates that written examination are inappropriate, and open the door for ambiguity and misinterpretation due to the variance in student backgrounds. Moreover, differentiation of examination formats for different students may not be feasible due to limited resources. Thus, an interactive strategy with ongoing feedback is the way to go.

We propose an assessment strategy that considers both in-class assignments and group projects. The final grade is a combination of a group grade (that we call *base grade*) and an individual grade based on: contract compliant, contribution reports, size of individual responsibilities, and invested hours. The group itself will autonomously aid in defining the individual grade based on member rankings of each others.

A base grade, on the other hand, is more involved as it depends on different factors. For instance, considerations to group's documentation, organisation, presentation quality, contribution to the in-class discussion, and successful project delivery are taken into account.

Our goal is to enforces a sort of equilibrium in the group. That is, no member has incentive to deviate from the goal of the group as otherwise it will be reflected in peer-evaluation. This can mitigate the problem re-grouping based on mutual interest and may help to enforce positive interdependence.

5 Concluding Remarks and Future Directions

In this paper, we discussed the problem of teaching theoretical computer science courses in terms of diversity of attending students. Our definition of diversity considered the variance in students skills, knowledge, and interests due to their different education paths. We explored the literature that consider similar issues such as differentiated instruction [46]. We explained how differentiated instruction can be hard to implement in practice due to limitation of resources. We, then, proposed a differentiated instruction model in a cooperative learning settings as a mitigation strategy. To showcase our developments and to explain the various concepts, we used a theoretical computer science course, named "Testing, Debugging, and Verification" as a running example. We used the latter to illustrate how a concrete implementation of our proposal can be realised in practice. Our proposal tackled two main challenges that may arise due to student's variance, namely group formation and efficient assessment.

For future work, we would like to implement this proposal for the above mentioned course and evaluate its effectiveness in practice. There are still few issues that need to be handed in a better way. For instance, the teaching team is usually made up of a teacher, examiner, and a number of TA students. The students are subject to change, due to graduation, long-term research visits, etc. Moreover, there are no pedagogic prerequisites for Phd students to start TA responsibilities. Thus, every time the team changes, there is a preparation time for the newcomers that may impact on the overall quality of the course. Another issue that need to be addressed is the student's response in terms of engagement, preparation, and satisfaction. Although this course might be better in terms of learning outcomes, it can also be very demanding for the students as more time out of class has to be invested.

References

1. Regulation for bachelor's theses - implementation and assessment c 2023-0357. `https://www.chalmers.se/api/media/?url=https://webbpublicering360.portal.chalmers.se/Extern/Home/Download?recordnor=904720%262023_03%261116748_1_1.PDF%26ex`, accessed: 2023-04-15
2. Syllabus: Testing, debugging and verification (tda567/dit084). `https://www.gu.se/en/study-gothenburg/testing-debugging-and-verification-dit084`, accessed: 2023-04-15
3. Adams, N.E.: Bloom's taxonomy of cognitive learning objectives. Journal of the Medical Library Association: JMLA **103**(3), 152 (2015)
4. Asok, D., Abirami, A., Angeline, N., Lavanya, R.: Active learning environment for achieving higher-order thinking skills in engineering education. In: 2016 IEEE 4th International Conference on MOOCs, Innovation and Technology in Education (MITE). pp. 47–53. IEEE (2016)

5. Bednarik, R.: Problem-based learning in teaching theoretical computer science. In: International Conference on Engineering and Research. pp. 801–807 (2004)
6. Biggs, J.: Aligning teaching for constructing learning. Higher Education Academy **1**(4), 1–4 (2003)
7. Bondie, R.S., Dahnke, C., Zusho, A.: How does changing "one-size-fits-all" to differentiated instruction affect teaching? Review of Research in Education **43**(1), 336–362 (2019)
8. Bonwell, C.C., Sutherland, T.E.: The active learning continuum: Choosing activities to engage students in the classroom. New directions for teaching and learning **1996**(67), 3–16 (1996)
9. Brown, A.L., et al.: Learning, remembering, and understanding. technical report no. 244. (1982)
10. Chamberlin, M., Powers, R.: The promise of differentiated instruction for enhancing the mathematical understandings of college students. Teaching Mathematics and Its Applications: An International Journal of the IMA **29**(3), 113–139 (2010)
11. Cohen, E.G.: Restructuring the classroom: Conditions for productive small groups. Review of educational research **64**(1), 1–35 (1994)
12. Conley, D.T., French, E.M.: Student ownership of learning as a key component of college readiness. American Behavioral Scientist **58**(8), 1018–1034 (2014)
13. Crider, A.: " hot seat" questioning: A technique to promote and evaluate student dialogue. Astronomy Education Review **3**(2), 137–147 (2004)
14. Dederichs, A.S., Karlshøj, J., Hertz, K.: Multidisciplinary teaching: Engineering course in advanced building design. Journal of Professional Issues in Engineering Education and Practice **137**(1), 12–19 (2011)
15. Dolmans, D.H., Loyens, S.M., Marcq, H., Gijbels, D.: Deep and surface learning in problem-based learning: a review of the literature. Advances in health sciences education **21**, 1087–1112 (2016)
16. Ernst, D.C., Hodge, A., Yoshinobu, S.: What is inquiry-based learning. Notices of the AMS **64**(6), 570–574 (2017)
17. Erol, M., Özcan, A.: Exemplary technology incorporated contemporary active learning environments for stem courses. The Eurasia Proceedings of Educational & Social Sciences **4**, 530–537 (2016)
18. Feldman, K., Denti, L.: High-access instruction: Practical strategies to increase active learning in diverse classrooms. Focus on Exceptional Children **36**(7) (2004)
19. Gehringer, E.: Active and collaborative learning strategies for teaching computing. In: 2007 Annual Conference & Exposition. pp. 12–167 (2007)
20. Gillies, R.M., Ashman, A.F.: The effects of cooperative learning on students with learning difficulties in the lower elementary school. The Journal of Special Education **34**(1), 19–27 (2000)
21. Gleason, B.L., Peeters, M.J., Resman-Targoff, B.H., Karr, S., McBane, S., Kelley, K., Thomas, T., Denetclaw, T.H.: An active-learning strategies primer for achieving ability-based educational outcomes. American journal of pharmaceutical education **75**(9) (2011)
22. Graham, C.R.: Blended learning systems. The handbook of blended learning: Global perspectives, local designs **1**, 3–21 (2006)
23. Hall, S.R., Waitz, I., Brodeur, D.R., Soderholm, D.H., Nasr, R.: Adoption of active learning in a lecture-based engineering class. In: 32nd Annual frontiers in education. vol. 1, pp. T2A–T2A. IEEE (2002)
24. Harahap, L.P.A., Mulyawati, Y., Sukmanasa, E.: The effect of applying the mentimeter media assisted problem based learning model on mathematics learning outcomes in data presentation materials. International Research-Based Education Journal **5**(1), 128–139 (2023)
25. Hussin, H., Jiea, P.Y., Rosly, R.N.R., Omar, S.R.: Integrated 21st century science, technology, engineering, mathematics (stem) education through robotics project-based learning. Humanities & Social Sciences Reviews **7**(2), 204–211 (2019)
26. Kaddoura, M.: Think pair share: A teaching learning strategy to enhance students' critical thinking. Educational Research Quarterly **36**(4), 3–24 (2013)
27. King, A.: From sage on the stage to guide on the side. College teaching **41**(1), 30–35 (1993)
28. Krishan, I., Al-rsa'i, M.: The effect of technology-oriented differentiated instruction on motivation to learn science. International Journal of Instruction **16**(1), 961–982 (2023)
29. Lindner, K.T., Schwab, S.: Differentiation and individualisation in inclusive education: a systematic review and narrative synthesis. International Journal of Inclusive Education pp. 1–21 (2020)
30. de Los Rios, I., Cazorla, A., Díaz-Puente, J.M., Yagüe, J.L.: Project–based learning in engineering higher education: two decades of teaching competences in real environments. Procedia-Social and Behavioral Sciences **2**(2), 1368–1378 (2010)
31. Marks, M., O'Connor, A.H.: The round-robin mock interview: Maximum learning in minimum time. Business Communication Quarterly **69**(3), 264–275 (2006)
32. Hernández-de Menéndez, M., Vallejo Guevara, A., Tudón Martínez, J.C., Hernández Alcántara, D., Morales-Menendez, R.: Active learning in engineering education. a review of fundamentals, best practices and experiences. International Journal on Interactive Design and Manufacturing (IJIDeM) **13**, 909–922 (2019)
33. Nicol, D., Thomson, A., Breslin, C.: Rethinking feedback practices in higher education: a peer review perspective. Assessment & Evaluation in Higher Education **39**(1), 102–122 (2014)
34. O'Connor, R.E.: Teaching word recognition: Effective strategies for students with learning difficulties. Guilford Publications (2014)
35. Piaget, J.: The origins of intelligence in children new york: International university press, 1952 (1936)
36. Pirker, J., Riffnaller-Schiefer, M., Gütl, C.: Motivational active learning: engaging university students in computer science education. In: Proceedings of the 2014 conference on Innovation & technology in computer science education. pp. 297–302 (2014)

37. Polo-Blanco, I., González López, E.M.: Teaching addition strategies to students with learning difficulties. Autism & Developmental Language Impairments **6**, 23969415211045324 (2021)
38. Ralph, R.A.: Post secondary project-based learning in science, technology, engineering and mathematics. Journal of Technology and Science Education **6**(1), 26–35 (2016)
39. Roberts, J.L., Inman, T.F.: Strategies for differentiating instruction: Best practices for the classroom. Taylor & Francis (2023)
40. Schmidt, H.G.: Foundations of problem-based learning: some explanatory notes. Medical education **27**(5), 422–432 (1993)
41. Senthamarai, S.: Interactive teaching strategies. Journal of Applied and Advanced Research **3**(1), S36–S38 (2018)
42. Sharma, B., Steward, B., Ong, S., Miguez, F.: Evaluation of teaching approach and student learning in a multidisciplinary sustainable engineering course. Journal of cleaner production **142**, 4032–4040 (2017)
43. Smith, C.V., Cardaciotto, L.: Is active learning like broccoli? student perceptions of active learning in large lecture classes. Journal of the Scholarship of Teaching and Learning **11**(1), 53–61 (2011)
44. Stager, A.: Differentiated instruction in mathematics. Ph.D. thesis, Caldwell College (2007)
45. Tomlinson, C.A.: Differentiated instruction. In: Fundamentals of gifted education, pp. 307–320. Routledge (2012)
46. Tomlinson, C.A., Brighton, C., Hertberg, H., Callahan, C.M., Moon, T.R., Brimijoin, K., Conover, L.A., Reynolds, T.: Differentiating instruction in response to student readiness, interest, and learning profile in academically diverse classrooms: A review of literature. Journal for the Education of the Gifted **27**(2-3), 119–145 (2003)
47. Tosic, P.T., Beeston, J.: Teaching theoretical computer science and mathematical techniques to diverse undergraduate student populations. In: 2018 ASEE Annual Conference & Exposition (2018)
48. Tseng, K.H., Chang, C.C., Lou, S.J., Chen, W.P.: Attitudes towards science, technology, engineering and mathematics (stem) in a project-based learning (pjbl) environment. International Journal of Technology and Design Education **23**, 87–102 (2013)
49. Tudge, J.: 6 vygotsky, the zone of proximal development, and peer collaboration: Implications for. Vygotsky and education: Instructional implications and applications of sociohistorical psychology p. 155 (1992)
50. Valiandes, S., Neophytou, L.: Teachers' professional development for differentiated instruction in mixed-ability classrooms: investigating the impact of a development program on teachers' professional learning and on students' achievement. Teacher Development **22**(1), 123–138 (2018)
51. Wittrock, M.C.: Generative processes of comprehension. Educational psychologist **24**(4), 345–376 (1989)
52. Witzel, B.S., Mercer, C.D., Miller, M.D.: Teaching algebra to students with learning difficulties: An investigation of an explicit instruction model. Learning Disabilities Research & Practice **18**(2), 121–131 (2003)
53. Zaretskii, V.: The zone of proximal development: What vygotsky did not have time to write. Journal of Russian & East European Psychology **47**(6), 70–93 (2009)