Can every student succeed in higher education? Introducing time differentiation and “follow-up” courses to a curriculum in Informatics

Anne Karin Wallace and Kai A. Olsen
Molde University College, Molde, Norway
anne.k.wallace@himolde.no

Abstract
In this paper we introduce a concept of time differentiation and follow-up courses, the idea being to let students choose their own learning curve. For example, by postponing a theoretical course a year by inserting a practical follow-up course before it, a weak student can get a better training and understanding of basic concepts before new theoretical topics are introduced. Good students can choose courses in their “normal” order, thus the cost of accommodating both groups are minor. These ideas have been tried at the Informatics program at Molde University College, Norway. We include some preliminary results based on two years experience with the new curriculum.

Introduction
Universal access to higher education sets new challenges to the institutions offering education at this level. While a university degree previously was for the few, the very best students, now a major part of students from high school enter universities. This is especially the case in countries, such as Norway, which provide free higher education. Here more than 50% enter a college or university. This implies that students will have very different qualifications and background. While the excellent students are still here, we also have to provide for students that have weak writing, foreign language or mathematical skills. Because of the size of this group we cannot longer avoid the problem by giving Fs and let the students drop out. This may cause severe financial problems for the institutions.

High schools met this problem earlier and have developed methods to get students through their program. This has mainly been achieved through offering students a large menu of programs that require quite different skills. This is not quite so easy to replicate within higher education, especially if one is in a situation that one have to accept every applicant. For each program, take Informatics as an example, there is a set of basic courses that students have to attend. This set can be defined formally by national or international professional agencies, or it can be defined more informally. However, still using informatics as an example, one will find topics as programming, algorithms, databases, networks and operating systems in this basic set. Thus, topic differentiation is limited, especially in the first (and perhaps most difficult years) in a study program.

Here we shall present our experiences from the Informatics program at Molde University College, Norway, where a system of time differentiation is used to accommodate students with different skills. This system is augmented with the introduction of a set of practically-oriented follow-up courses. A follow-up course will have much the same agenda as a previous more theoretically-oriented course, but the idea is now to practice the topics introduced earlier. The idea is to let students follow the same basic courses, but to let each student choose her own timing, i.e., when to take a course. The follow-up courses will help students in establishing a solid platform of competence, before moving to new topics.
Initially, we have applied these methods to our teaching in computer programming. The new system has been received very well by a large and diverse group of students. The results indicate improved understanding of computer programming among weak students.

**Universal access to education**

After the reform of upper secondary education (Reform 94) in 1994, everybody between 16 and 19 have a statutory right to attend three years of upper secondary education (high school). This right is used by nearly 100% of the age group, while approx 50% attend lines of study which prepare for higher education. During the last 50 years the number of students in Norwegian colleges and universities has increased from 5000 to 170000 (NOU 2000:14). We see the same trend in most industrialized countries (Altbach 1992). Differentiation is seen as important to accommodate new groups of students, for example in the report from the British National Committee for Inquiry into Higher Education (Dearing 1997). In the report differentiation is mentioned in the following contexts:

- Lifelong learning, the need for people to take higher education during their working lifetime.
- Distance learning, the need for people to attend the programs they want delivered in a flexible way.
- International recognition of qualifications, getting credits for higher education from another country.
- Modularization, being accredited for shorter studies, and maybe build up a degree over long time.
- Quality of teaching.

The Dearing Report builds on a view upon learning which puts the student at the centre of the learning and teaching process:

> Great teachers create a common ground of intellectual commitment. They stimulate active, not passive, learning and encourage students to be critical, creative thinkers, with the capacity to go on learning after their college days are over (Boyer 1990)

A similar view upon teaching is reflected in the educational reforms in Norway: reform of upper secondary education (1994); reform of primary education (1997) and the quality reform (higher education, 2000). The Quality Reform, a reform in higher education, based on the White Paper (St.meld. nr. 27, 2000-2001) entitled ‘Do your duty - demand your rights’ is implemented from the fall semester 2003. One of the focuses in the reform is that the student should succeed with his/her education. Curriculum changes, improved teaching methods and improved organizing of the topics taught is recommended. Instead of treating the students as a mass an individualized follow up of each student is recommended. This is to be implemented by personal advice, more student-active ways of teaching, better relation between assessment and teaching and by enforcing individual plan of studies. The student’s role as a person working actively with learning is also emphasized in the reform.

However, little is said about the great variation in skill and background of students. It may not be politically correct to talk about weak students with low motivation, but these are a part of every class. We are in doubt if better advice and more tutoring are the remedies needed to get these students though the program. From our experience we see that students are not good at taking the advice offered by faculty. Weak students often have an extraordinary belief in their own skills, even if proven otherwise by previous courses. Tutoring may help, but we have seen negative effects where students expect to be tutored out of complicated problems, instead of handling these on their own. As an additional measure we therefore want to focus on differentiation.
Differentiation - from high school to higher education

In Norway, after 1994, with almost all the 16 to 19 year olds in upper secondary school, the problem of differentiation has been an important task in this level of education. In 1999 the project “Differentiation and adaptation in upper secondary school” was started. All the upper secondary schools in the country participated. They were free to choose their own way of facing the differentiation problem, and a lot of different solutions have been tried. Different ways of organizing the school day and year, different ways of grouping the students, different ways of organizing teaching and guidance, new student-active ways of learning, and ICT are among the things that have been tried. The project ended in 2003 and have been evaluated by a research group (Berthinussen et al. 2001, Dale et al. 2001 – 2003).

The evaluation points out seven principal categories which are considered important when working with differentiation:

1) The students’ background. Achieve knowledge about the students’ qualities and abilities.
2) Knowledge of the curriculum. Make the curriculum known to the student and tied to work plans.
3) Individual task and time plans. This means everybody will not reach the same goals and they do not spend the same time on reaching the goals for their learning.
4) How teaching is organized. Organize the students’ working-day, teaching and learning activities (weekly schedules, schedules for a semester etc.).
5) Learning arenas. Use various learning arenas (classroom, laboratory, library, work experience, ICT etc.).
6) Teaching methods. Use various work methods and learning activities (collaborative work, ICT, lecture, project, PBL).
7) Evaluation. Continuously evaluate the work done on differentiation.

Every project carried out in an upper secondary school has touched one or more of these categories. The four points below summarize some of the results from the projects:

• Different ways of organizing teaching and different teaching methods have been tried, but there is not a clear winner.
• Students who know the curriculum well are more motivated for learning.
• Variation in the ways of teaching makes the students more active in the learning process.
• Students participating actively in the education are necessary for success.

The reason for focusing on differentiation is to get every student to work with learning, to be active in the process of learning. The evaluation of the projects shows that there is no simple formula for achieving this goal. Several reasons for inactivity are reported by teachers:

• Some students spend their time on part time working and leisure activities; they are not spending time on studying.
• Some students have mental sufferings or social problems which prevent them from studying.
• Some students come with unrealistic expectations; they expect to get good marks without working, without showing initiative and participation.
• Some students think that when they by law have the right to go to school they also have the right to get a good mark.

We will comment shortly on the seven categories’ relevance in higher education.
1:  *The students’ background.* Formally we know the background of the students who are admitted into higher education, in practice we know that their qualifications vary a lot. In higher education it is considered the students’ responsibility to attend courses/studies for which they have the required background. The responsibility of the school is to tell them what this required background is. Probably the institutions still have to get more involved at this point, either by more guidance or by applying more specific entrance requirement to studies/courses.

2:  *Knowledge of the curriculum.* The motivation from knowing the reason for working with an activity is probably just as important in higher education as in upper secondary school. The teacher will mostly be responsible for this point at both levels.

3:  *Individual task and time plans* are, at undergraduate level, mostly implemented by the principle of modularization: The students can more or less freely compose their plan of studies containing a number of courses. In higher education, the curricula are often made by the individual school, maybe by an individual teacher. This opens for the possibility of differentiation by making courses which fit the qualifications and needs of different student groups. We think the students can benefit from this kind of differentiation if the diversity in student needs and qualifications is considered when curricula are made. This way of differentiation can only be used to a limited extent in studies where there are specific standards the graduates must meet, like in nursing schools.

4 and 5:  *How teaching is organized* and the use of different *Learning arenas*. Organizing the teaching in different ways, and using various learning arenas, are probably easier and more common in higher education than in upper secondary school. Buildings and equipment are better, and there are often no fixed class-structures or national exams.

6:  *Teaching methods.* The two last categories are focused in the Quality reform, and all institutions offering higher education are focusing on applying a variety of teaching methods and learning activities including ways off assessment.

7:  *Evaluation.* Student-evaluation of teaching is common in higher education, and is recommended improved in the Quality reform. It is common for the institutions to evaluate the curricula and teaching. These evaluations should also include evaluation of differentiation.

The seven categories seem to be covered by modularization and quality of teaching which are emphasized in the Dearing report. The categories seem relevant for higher education. There are important differences in the way education is organized on the two levels. In upper secondary school attending classes is mandatory, the students are organized in classes, the curricula are national, and there are national exams. The lack of most of these restrictions in addition to better buildings and equipment in higher education makes organizing the teaching and using different learning arenas easier to deal with. *Student background* and *individual task and time plans* cannot be handled in the same manner in the two levels of education. The competence among teachers on didactics is probably higher in upper secondary school than in higher education, this makes *Teaching methods* more of a challenge for higher education. *Students’ knowledge of the curriculum* is an issue for teachers at both levels. The importance of focusing on student activity and participation is probably not restricted to an age group, and the reasons for student inactivity reported from secondary school are well known in higher education as well.

**Time differentiation and practically oriented courses.**

In Molde University College, we have had a large and diverse student group in the IT department for some years. In 2001, a curriculum change was made to try coping with this. The changes are
related to category 3 and 6: Diversity in curriculum content and teaching methods. The experiences with this change are reported below.

The program for the Bachelor degree in Informatics consists of courses giving 6-18 credits. There are some compulsory courses. The rest can be chosen from the courses offered by the college, but a minimum of 108 out of 180 credits must be from the IT-department. We publish a number of suggested plans of study, but the students are mostly free to choose courses in the order they want (some courses require others taken in advance). The courses are offered with the following goals:

- Offer courses needed for further studies (masters degree in IT or logistics)
- Offer courses with a content relevant for work in local commerce, industry and public sector
- Offer courses reflecting new technology and new topics relevant for people who work with IT

Knowledge of computer programming is considered essential for everybody working with IT. Courses of computer programming are therefore found in every IT-curriculum. At the same time a considerable amount of IT-students have problems getting satisfactory results in these courses. In Molde all of the students will, in their first semester, attend the course *Introduction to programming* where they are introduced to basic concepts of computer programming. In the second semester they attend the course *Advanced programming* where they learn a new language (Java) and focus more on object-oriented programming. The students who had problems in the first course seemed to get even more problems in the next one. To cope with this we introduced the following changes to the curriculum:

- Differentiation with respect to when a course can be attended
- Introduction of two new practically oriented courses

The students who perform badly on the introductory course are now advised to delay the advanced programming course from the second to the in fourth semester. They are advised to attend the course *Programming practice* in their second semester to improve the understanding of the concepts of programming. This follow-up course introduces few new topics. Instead students have to perform a large project, utilizing the ideas introduced in the introductory course.

![Progression](image)

*Figure 1. Time differentiation*

This idea is illustrated in Figure 1. Normal progress will be to take A (*Introduction to programming*), then C (*Advanced programming*). We now offer the option of letting students delay C for a year, replacing C in the second term by B (*Programming practice*).
This may imply that they will not be able to attend all of the more advanced courses, such as *Data structures and algorithms* during the standard three years. However, these courses will anyway be of little attraction to the weaker students.

**Programming practice.**

*Programming practice* is a course which focuses on using the knowledge from the introductory course. Most of the time is spent working in small groups on a project, made as realistic as possible. There are some lectures to introduce a few new topics. Those are mostly introduced by presenting examples that are accessible afterwards for the students via the course web site. Teaching assistants are available three days a week to answer questions and give guidance. The students have to hand in their work five times during the semester, and the teacher or the teaching assistant give feedback each time. We have chosen to work on an e-commerce project. This means a lot of programming where students practice things they have seen the previous semester, but also learn some new concepts and technology (programming for the Web). The workload in the course is supposed be the same as in *Advanced programming*, but fewer new topics are introduced. This is supposed to give more time to reflect on and get comfortable with the basic principles of programming. When the students attend *Advanced programming* in their fourth semester they are hopefully better prepared for this course.

*Programming practice* has been offered twice, in 2002 and 2003. We have studied which students attended the course, and their results. We have also followed the students from the introductory programming course these two years to see what course they choose afterwards and what their results are. In addition, the students in *Programming practice* have answered a questionnaire to give us some feedback on their learning activities and motivation for attending the course.

**Time differentiation.**

The composition of the student group attending the course has not been what we expected. Out of the 113 students who have attended the course, 41 have been in their first year of studies. Out of the 41, 13 attended *Advanced programming* in parallel. This means that at most 28 of the students who attended the course were in the target group.

In the questionnaire for 2003 we listed many possible reasons to attend the course, and asked the student to mark those that were relevant for her/him. The result is shown in Figure 2.
Comparing the marks in *Introduction to programming* with marks in *Advanced programming* shows that a lot of students have problems with programming. The comparison is done by grouping the marks in the two subjects into three groups, and counting the number of students in each group. The result is shown in Figure 3. The number of students who have moved to a lower group in the second course is in bold. The table contains students from 2002 and 2003 who attended *Advanced programming* in their second semester, i.e., according to the normal schedule. Those who did not get good results in the first course got even worse grades in the second. The number of students who fail is low in both courses, but many obtain poor results.

<table>
<thead>
<tr>
<th>Number of students</th>
<th>Advanced programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to programming</td>
<td>Best</td>
</tr>
<tr>
<td>Best (1.0 – 1.9)</td>
<td>16</td>
</tr>
<tr>
<td>Medium (2.0 – 2.9)</td>
<td>1</td>
</tr>
<tr>
<td>Weak (3.0 – fail)</td>
<td>0</td>
</tr>
</tbody>
</table>

This table also shows that many students do not follow our advice to postpone *Advanced programming* until the fourth semester. This is what we have seen from other situations. Students have a high opinion on their own abilities, and choose to go directly to the advanced course in order not to miss the opportunity to take even more advanced courses later. We see two possible actions here. One is a closer follow-up of the students. This can be done using the students’ individual plan of studies. The student is supposed to compose this plan after consultation with a
teacher. The teacher can use this situation to help the student make realistic goals for his/her studies. The other solution is not allowing students with marks below a certain level to enter the advanced programming course. The best solution is perhaps to try both alternatives, set strict requirements and then motivate students for the alternatives.

In the table below (Figure 4) we show what happened to students who followed our advice and attended Advanced programming in their fourth semester. These students are only found in the 2003-group. Their results in Programming practice showed an improvement compared to the introductory course.

<table>
<thead>
<tr>
<th>Introduction to programming</th>
<th>Advanced programming</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best</td>
</tr>
<tr>
<td>Best (1.0 – 1.9)</td>
<td>1</td>
</tr>
<tr>
<td>Medium (2.0 – 2.9)</td>
<td>1</td>
</tr>
<tr>
<td>Weak (3.0 – fail)</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 4. Comparing marks: Introduction to programming compared to Advanced programming for students with Programming practice. Bold numbers indicate improvement.

Up to now we have only twelve students who have continued with Advanced programming in semester four after attending Programming practice. There seems to be an improvement among the weak students, but unfortunately we have too few data to confirm the results.

Learning activities related to practically oriented courses.

In the questionnaire in 2003, the students were asked how important the different learning activities are for them. This was answered by giving the activities marks on a scale from not important at all to very important.

![Figure 5. Learning activities regarded important for learning.](image)

Figure 5 shows the number of students who consider the different activities as very important. Guidance, collaboration, working on a large project, and regular hand-ins obtained the highest scores. These activities were focused on in the course, so the result is not unexpected.
Surveys in other programming courses show that the students usually consider working with exercises and attending lectures the most important learning activities. Our results show that the students think they benefit from a variety of learning activities, so variation seems important. This corresponds to differentiation-category 6: Teaching methods. Even though the majority of the students like the learning activities used in a practically oriented course, there seems to be a few who have strong preferences towards lecture-oriented teaching.

**Motivation**

In the questionnaire, the students were requested to write a few bad and a few good things about the course. The positive comments were most interesting. One third of the respondents in 2002 wrote things like “Interesting, I can be creative, I learn a lot”. These kinds of comments are also found in questionnaires in other programming courses, for instance *Introduction to programming*, but with considerably lower frequency. Problem solving and creativity seems to be important motivation factors in practically oriented courses.

Comments like “Practice working on a large project”, ”Practice programming something ordered by others” and ”Realistic exercise” show that relevance for future work is also very important for motivation. Figure 2 indicates the same; the students attend *Programming practice* to learn the programming language Visual Basic and the .NET technology. Visual Basic and the .NET technology are used in companies where the students may work in the future.

**Conclusion**

We have introduced a concept of time differentiation and practical follow-up courses to a Bachelor program in Informatics. The idea is to accommodate different groups of students without having to develop separate curriculums, i.e., to introduce as few additional courses as possible. The preliminary results are promising. The follow-up courses tend to consolidate the topics introduced in the earlier, more theoretical courses, and give students a better background for taking the more advanced courses. However, it is a problem that some of the weak students do not follow our advice to postpone the next advanced course. Instead they try to follow the “normal” schedule, with poor results.

An unexpected high number of students attended the follow-up course, *Programming practice*. They achieved good marks and expressed satisfaction with the course. This supports the importance of using a diversity of teaching methods and learning activities. The practical approach taken in the course and the project’s relevance to future work were also appreciated. These principles can be further developed in other courses.

**References.**


Boyer, E.: Scholarship reconsidered, New Jersey, Carnegie Foundation for the Advancement of Teaching, 1990


Dale, E.L., Wærness, J.I., Andresen, R: Differensiering i et positivt selvforsterkende løp. 4. underveisrapport til evaluering av prosjektet ”Differensiering og tilrettelegging i videregående opplæring”, Oslo, Læringslabben, 2002


