Using Blended Learning to Differentiate the Learning Process in a Computer Programming Course

Olav Dæhli
University College of Southeast Norway (NORWAY)

Abstract

Teaching freshmen in an engineering education can be challenging due to different levels of skills, interests, knowledge, and prior experience among the students. The engineering education at the University College of Southeast Norway (USN), campus Porsgrunn, offers several paths to admission. For instance, approximately 60% of the students have a certificate of apprenticeship. Because of this, it is difficult to create one course design that fits all students.

This study focuses on a course named “Object Oriented Programming and Databases”. During the last three years, this course has been under major reconstruction. This provided an opportunity to review both the course content and the pedagogical approach to create a renewed course with new course material.

In order to adjust to different needs, it was of interest to develop a flexible study model. A way to accomplish that was to use blended learning. Blended learning is characterized as a combination of traditional classroom activities and online media resources. Many courses today will meet these criteria, but it can be challenging to find the optimal blend of face-to-face and online activities.

A socio-constructivist learning approach is assumed. Students are expected to build competence by engaging actively in team work and other learning activities facilitated by professors. The course consists of lectures and exercises, as well as individual and team-based work requirements. A key component is a student project implemented as a project-based learning element.

To be able to challenge students at different levels and with various professional interests, a large amount of learning items is developed, including many videos. Students are also guided to additional resources, like e-books and other online material. Some of these resources are basic and aimed at students that find it hard to learn programming, while others aim at students who want to expand their skills beyond what’s included in the curriculum.

To ensure a good blending of course elements, the course has undergone frequent evaluations in the last three years. Feedback from students is used as a basis for new adaptations. In 2016 the students’ motivation and work efforts were reported as high during the entire course. Many online resources were offered, but most of the students still attended the voluntary campus work. Students did very well on their projects. Videos were widely pointed out as being of particular importance. The students’ final grades were very good, except for a small group that failed. Possible reasons for failing are discussed, as is the value of different learning elements and course resources used throughout the course. Some suggested improvements are to include even more flipped classroom-thinking and more videos related to especially basic items.

Keywords: Engineering Education, Problem-based learning, Object-oriented programming, PBL, Flipped Classroom.

1 Introduction

This study is about the use of blended learning in a course at the engineering education at University College of Southeast Norway – Faculty of Technology, Natural Sciences and Maritime Sciences (USN – TNM), within a study program called “Computer Science and Industrial Automation”. The course, ‘Object-Oriented Programming and Databases’, is a 1st semester 10 ECTS course (European Credit Transfer and Accumulation System). The course has approximately 90 attendees each year.

The aim of the course is to teach students how to program event-driven applications with an object-oriented language, together with database modeling and implementation. It is considered to be a challenging course which requires continuous follow up from the students. If a student starts lagging behind, it is typically difficult to catch up later.
At the beginning of the course, there is usually a wide range in students' previous experience with programming. For example, there is a difference in the type of upper secondary school experience students have. While approximately 40% of students have a background in general studies and have only recently finished school, 60% of them have gone through vocational training, typically followed by a certificate of apprenticeship, and often several years of work experience. Some students have even taken an IT-emphasis during upper secondary school. Some might have had programming as a hobby, whilst still others may not have been doing any programming at all. Programming skills are not an admission requirement.

Those different preconditions make it critical to build an educational framework that can be adjusted to students with different abilities, aspirations and previous knowledge. If this is not taken into account when designing the course, it will put many students in a difficult situation. As Kierkegaard said: ‘If one is truly to succeed in leading a person to a specific place, one must first and foremost take care to find him where he is and begin there’ [1].

In the period of 2013 – 2016, the course was upgraded substantially. Since both the programming language and the database tools needed to be replaced, the course material also had to be replaced to fit the new environment. This presented an opportunity to overhaul the entire course, including the course material. Important aspects were to

- let course content and tools harmonize better with advanced courses following this course
- make the course flexible regarding the time and place to study
- adjust the course material to suit the different levels of preconditions

Instead of patterning the course content after an existing textbook, new customized course material was developed. Blended learning seemed to be a natural approach. There have been many definitions of blended learning since access to the World Wide Web was introduced. In The handbook of blended Learning [2] from 2006, different elements related to blended learning are discussed.

Here, blended learning is considered as “... learning experiences that combine face-to-face and online instruction.” [3]. These two elements are to be combined into one design, that forms a unified whole.

To encourage students to use the blended resources, the resources are designed to view the theory from different viewpoints. For many topics, a short introduction video is made available on Fronter, the Learning Management System (LMS). The compendium provides a more thorough explanation of the same topics. Lectures are used mostly for hands-on programming and explanations. Exercises have a progression in the degree of difficulty, to challenge all students on their own level. The different resources use different examples. Fig. 1 illustrates the different views to the same topic.

![Image of diagram](image-url)

**Figure 1. Each topic is illustrated in various ways on campus and online.**

The focus of this study is on the use of blended learning, where aspects as motivation, workload, educational resources and grades are considered.

Sections 2 to 4 describe the course and the course design, the pedagogical approach and the research methods. Results and discussion are presented in section 5 and a conclusion in section 6.
2 DESCRIPTION OF THE COURSE AND COURSE DESIGN

The programming course is using C# as programming language and Visual Studio as IDE (Integrated Development Environment). In addition, the course includes database theory, modeling and implementation. An application named Erwin is used for modeling.

Programming and databases are often separated into different courses at universities, but here they are integrated in one. Students seem to be getting a better understanding of both programming and databases when they learn how to practically implement a database as part of their solution. Here is a general description of the course elements in the order they are taught:

1. Programming fundamentals (C#)
2. Database fundamentals, including SQL (Structured Query Language)
3. Modeling a database with ERD (Entity Relationship Diagram)
4. Combining the above elements to create an application implementing a database
5. More about programming concepts related to object oriented programming

A weekly progress plan is used as the front page for the course in the LMS, so that students always have an overview of the entire course. Each course date has a description of topics and links to relevant resources for preparations and homework. The schedule contains links to lecture notes, files to download as preparation for lectures and exercises, videos for preparation and homework, solutions to exercises (pdf-files or videos), and other relevant resources.

The row showing the upcoming lesson is always highlighted in yellow. One example of such a row is shown in Fig. 2. Icons are used to make it easy to see if a link is to a video, zip-file, pdf-file etc.

<table>
<thead>
<tr>
<th>Week/Day</th>
<th>Subject</th>
<th>Lecture-slides</th>
<th>GUI and other files for use in lectures and exercises</th>
<th>Literature and videos</th>
<th>Exercises</th>
<th>Suggested solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 39:</td>
<td>Databases: Queries against joined tables</td>
<td>Slides-Lecture 3</td>
<td>• zip-file with five scripts to be used in the lectures (MovieDatabase 3)</td>
<td>Database-book: Chap. 7 - 8.6 (p 34-48)</td>
<td>• Exercise 3 (Databases)</td>
<td>Suggested solutions</td>
</tr>
<tr>
<td>Thurs 29/9</td>
<td></td>
<td></td>
<td>• zip-file with three scripts that import data to the tables showed in the videos. (zip-fil)</td>
<td>Video to be seen before lectures:</td>
<td>• Exercise 3 (Databases)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• zip-file with four scripts for exercises</td>
<td>SQL-queries against joined tables (ca. 23 minutes)</td>
<td>• Exercise 3 (Databases)</td>
<td>Video-solutions</td>
</tr>
</tbody>
</table>

Figure 2. Two weeks scheduled with preparation elements and other resources.

Having the complete overview/schedule showing up each time a student enters the virtual course room has proven to be a simple, yet effective, method to help students stay on track with the course. It ensures that students always have a good overview of their own progress as well as the entire course.

Some lectures are of the more traditional kind, with theory presented by using PowerPoint-slides, but most of the lectures are hands-on programming with students working on their laptops. Usually students have to download some C#-files containing Graphical User Interfaces (GUI) etc. in advance. Such files are provided as starting points for lectures and exercises throughout the semester, so that the focus of the lecture can be on the relevant aspects for that week’s theory.

3 PEDAGOGICAL APPROACH

To achieve good blended learning, the physical and the virtual learning environment need to support each other.
1 Physical environments: Classrooms, seminar rooms, cantina and other relevant workplaces
2 Virtual environments: Virtual class rooms, virtual team rooms and virtual collaboration tools

These environments should be conceptualized as one unified learning environment. Action Research is one way to achieve a good blending of course elements, by continuously making small adjustments, getting feedback and then making adjustments based on the feedback.

The course covered in this paper is pedagogically based on a socio-constructivist approach. Construction of knowledge is assumed to take place when students engage in learning activities in collaboration with other students. Illeris illustrates learning as a process where cognition and psychodynamic dimensions always influence each other, and where both are part of an interaction with a social dimension. See Fig. 3. Learning takes place in the tension between these dimensions [4].

Teams are utilized in this course to establish social learning communities. Meaning is considered to be created through an ongoing negotiation between individuals interacting in the social environment [5].

---

USN – TNM, campus Porsgrunn, has since 1982 based many of its engineering courses on a project-based learning (PBL) approach [6]. Elements of PBL are also implemented in this course. Through work requirements, students work collaboratively in teams towards a common goal.

Today’s technology provides many tools to collaborate online. At USN-TNM, a Learning Management System (LMS) named Fronter is used as the main platform for supplying course material, virtual rooms and collaboration tools. In addition, the campus provides good physical facilities of which the students make good use of.

Creating a virtual environment is not only about offering a wide range of online material, such as videos and other digital resources, but also about giving the students the necessary tools to inspire them to take ownership of their own learning. Each team is given ownership of a virtual room in Fronter to collaborate by sharing documents, having discussions and more. They can also invite others.

Students are also recommended to participate in online forums on the internet. Within the area of programming and databases, there are large online communities with Frequently Asked Questions (FAQs), forums and other resources. It is better for students to get accustomed to search for answers and participate in these kinds of forums, than to feed them with direct answers to all their questions. This is also one of the learning aims of PBL.

It is important to create environments were all students have to contribute and to avoid passive observers. To ensure this, each team member must work on an individual work requirement the first part of the semester. Each student chooses the kind of application they wish to program. A check list with requirements ensures that important learning aspects are included in the solutions. The students work individually but are encouraged to collaborate within the team.

The last part of the semester, the teams are given a work requirement to be solved as a team task. Now they have to select a task from a set of given alternatives, but with a high degree of freedom regarding how they want to solve it. Programming and databases are then combined into one solution. A typical workflow for a students’ weekly work at a general level is illustrated in Fig. 4. The work includes both campus and online activities.
The online part of blended learning is not meant as a replacement for professors. Instead, campus activities can now be targeted towards providing the students with a deeper understanding, for instance through guidance adapted to the individual student.

4 METHODOLOGY

The research was conducted as Activity Research, where different changes to the course were introduced over a period of three years. Quantitative and qualitative methods are used to answer the research questions. These methods are described in the sections below.

4.1 Surveys (Close ended and open ended surveys)

Surveys have been conducted every semester. The respondents have answered anonymously, so they could feel free to give any feedback they wanted. The surveys consist mostly of close-ended questions with multiple choice and multiple answers, so responses are easy to quantify. A number of open-ended anonymous survey questions is also included to give the students an opportunity to speak freely and perhaps reveal relevant issues that the questionnaire did not address.

4.2 Evaluation meetings with class representatives

Evaluation meetings are arranged with student representatives from each field of study every year. The class representatives meet with their classes to perform an evaluation without the instructor present. After this meeting, the representatives meet their professors, to pass on the students' opinion of what went well and what could be improved.

4.3 Statistics

Statistics from the LMS are observed to get an impression of the use of videos and other resources. Students' grades are studied over several years.

5 RESULTS AND DISCUSSION

In this section results are presented and discussed in the light of the research questions.

5.1 Mapping students' prior knowledge

Many students say they feel that other students have more prior programming experience than themselves at the start of the semester. To tailor the blended learning model to the different needs of the students, a mapping of the students' prior computer programming was performed at course start in the autumn of 2016.

The result in Fig. 5 shows that most students did not have any prior programming experience. This was a 'multiple answer' question, so respondents could check for more than one alternative. The survey was answered by 73 out of 82 students, where 38 students (46 %) replied they did not have any programming experience at all. The answers from the rest (35 students) were distributed among the various options, but mainly on Python and other languages.
Supplementary open-ended questions reveal that the knowledge of Python mainly relates to a two days voluntarily course offered new students by the faculty before Study Start. The answer ‘Other programming languages’ is mainly selected by students who have worked as an electrician or automation technician. Some of these students have been working with Programmable Logic Controllers (PLC), but normally not common programming languages.

According to these results, the impression of fellow students being more experienced seems to be more of a feeling than a fact. Nevertheless, it is important to help students overcome this concern. Videos that guide students through basic theory are one measure to help students overcome this ‘fear’. Letting students collaborate in teams, virtually and face-to-face on campus, is another, because this allows them to draw on one another’s “expertise”. Campus-activities where students can request help from supervisors and learning assistants are also considered important.

5.2 Motivational aspects

The course is mandatory for all students studying “Computer Science and Automation” (CSA) and “Electrical Power Engineering” (EPE). Open-ended anonymous survey questions have revealed previously that it is more problematic to motivate the EPE students, because they do not always see the same relevance of basic computer programming as the CSA-students.

In a final evaluation before the course overhaul in autumn 2012, comments like “I feel this is not relevant to EPE-students, and should perhaps be an elective” or “As an EPE-student, this is not relevant to my education”, indicated this. This was also discussed in oral evaluation meetings. There were no such comments in 2016. In 2012, many students also found it hard to see the connection between lectured theory and practice. Other students reported that the pace of presenting the theory was too high and that they found it difficult to see the relationships between the different subjects presented in the course.

USN – TNM considers computer programming as an important course for both CSA and EPE-students, and wanted better learning outcomes and a better course design. Despite some negative comments in autumn 2012, other feedback that same year was positive, like “This is exciting, because I have not been programming before”, “Good teacher, good exercises”, “Interesting course, good teacher”.

A great difference in preconditions and interests makes it important to construct a course that is motivating to different kinds of students. Some motivational actions taken during the first weeks of 2016:

- A motivational video that illustrates what programming is, was introduced to first lesson
- Instructional videos explaining some basic programming elements were published in the LMS
- Students were presented with solutions developed by former students
- Students started early to create simple programs with a graphical user interface
In a survey after approximately two weeks in 2016, 93% of the students responded 4 or 5 on a scale from 1 to 5 on motivation. The results are presented in Fig. 6, where the left (dark grey) columns in the ‘clustered column chart’ show the motivation at course startup and the right (light grey) columns show the motivation halfway through the course. A number of 73 out of 82 students answered the survey.

Figure 6. Students’ motivation 2016 at the beginning of the course and halfway.

It is one thing to motivate students at the beginning of the course, but it is even more important to keep the students motivated through the entire course. As stated earlier, this is the kind of course that many students usually find hard. They have to learn a lot of new applications and tools and have to do a lot of logical, creative thinking to create good algorithms to implement in their applications.

It is expected that the motivation drops somewhat throughout the course when the programming becomes more difficult. According to Fig. 6, the motivation drops, but is still fairly high halfway in the course. Fig. 7 shows good scores for the degree of overall satisfaction with teaching (columns to the left) and curriculum (columns to the right) after the final exam.

Figure 7. Overall satisfaction with teaching and curriculum after the final exam 2016.

Here are some examples of anonymous comments (translated by the author) related to motivational aspects. These comments are by the author considered to be representative for many students:

- ‘Good videos with suggested solutions are frequently published’
- ‘Good exercises throughout the semester and it is especially helpful with videos showing solutions to exercises and theory from lectures’
• ‘Very good lecturer. Good lecture notes (compendium) and videos!’
• ‘Well-prepared lessons and good exercises’
• ‘Very good information in Fronter (the LMS). Awesome videos!’
• ‘Good compendiums. Competent supervisors and lecturers’
• ‘The work requirements gave me a good learning outcome before the exam and examples used in lectures were well prepared’
• ‘It was very clever of the professor to publish many resources in Fronter, for instance videos, which were of great use to me. This makes it possible for students to learn at their own pace’

The essence of comments related to suggested improvements:
• Many students did not like PowerPoint presentations presented in lectures. They rather prefer lectures where explanation and learning takes place through hands-on programming.
• Many students experienced trouble when installing software, due to hardware issues etc. Not having installed the software makes it difficult for students to keep up with the course progress.
• There are many requests from students for more videos.
• Students new to programming are dealing with many new words, concepts and terms, which can be confusing. They need help to get a clearer understanding of these early in the course.
• Several students ask for recordings of lectures in case they cannot attend for some reason or because they want to be able to see certain explanations again.

Some suggested solutions: The theory explained through ‘PowerPoint’ slides can be presented in videos instead. Several subjects are already covered this way, with good feedback from students. This would be a further move in the direction of a flipped classroom. Learning assistants could be hired to help students solve installation problems. The IT helpdesk is not familiar with these applications, so students have to get help from others. The request for more videos is mostly a resource issue, because it is time-consuming to produce videos of good quality. However, many videos can be reused, so the resources that need to be allocated would be less each year. Quizzes or Kahoots (game based learning tool) can possibly be used as additional learning resources to learn new words and terms.

Another indication of motivation is that 85-90% say they intend to attend lectures and exercises on campus, even though attendance is voluntarily and a great amount of virtual resources are available. Most of the students attended most of the lectures.

Attendance has also been high on exercises, but by natural causes attendance fell somewhat throughout the semester: After a while, many of the students become more self-driven and prefer to use team rooms where they collaborate with other team members rather than coming to class exercises. Suggested solutions available on videos also reduces the need for guidance. Throughout the semester, more and more of the questions are related to the students’ work requirements.

5.3 Workload considerations

Fig. 8 shows how much effort the students intend to put into this course (the dark grey columns). The same chart shows the weekly workload they report halfway in the course (the light grey columns).

Most students were prepared to invest much time and effort into the course. For a course of 10 ECTS, the norm of expected workload is 12-15 hours per week. The majority of the students expected to spend 10-17 hours weekly, which also is an indication of highly motivated students. The midterm evaluations indicate that the real workload has been even higher.
5.4 Blended activities and resources considered important by students

In a multi-answer questionnaire halfway through the course, students reported which educational offerings they considered as essential to their learning during this course. The result is shown in Fig. 9.

An interesting aspect is that campus lectures achieve the highest score on educational offerings essential to learning. More than 90% report this as important. The same applies to another course (ICT-tools) the same semester, where all students regardless of study discipline participated. 80% said they wanted videos as an addition to the lectures, not as a replacement. They also stated they consider lectures as an important element in their learning progress. The ICT-tools data is reported in a paper by O Dæhli [7]. This indicates that students consider blended learning to be a good solution.

![Figure 8. Expected and actual weekly workload reported by the students 2016.](image)

![Figure 9. Activities and resources considered essential to learning by students in 2016.](image)
5.5 Student Performance

The chart to the left in Fig. 10 shows the grades achieved in the autumn semester 2016. The chart to the right in Fig. 10 presents the number of A's and F's achieved during the period 2012 to 2016. Until 2014 it was approximately the same number of A's and F's, but the two last years, more A's than F's have been given. The results are presented as percentages, since the number of students has varied in the range 80 – 90 during the period.

![Graph showing grades and percentage of A's and F's from 2012 to 2016.]

Compared to earlier years, students in the autumn 2016 semester achieved better average grades than students from earlier years. See Table 1. The number 0 corresponds to F or Failed, and 5 to an A.

**Table 1. Average grades over the period 2012 to 2016.**

<table>
<thead>
<tr>
<th>Year</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
<th>2015</th>
<th>2016</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average grade</td>
<td>2.4</td>
<td>2.4</td>
<td>2.7</td>
<td>2.5</td>
<td>3.0</td>
</tr>
</tbody>
</table>

In 2016, eight candidates received a failing grade. A closer look at the grades from these students revealed an average grade of respectively 0.5 and 1.6 in two other courses they attended this semester, a math/technical course and a course in Project Methodology, ICT and Financial tools. These students seem to be academically weak students who would likely have problems with many courses within the engineering disciplines.

6 CONCLUSIONS

This is a course that blends online resources and traditional learning resources. Many changes to the course have been made since 2012. The actions taken have led to a course where most of the students are motivated throughout the entire course and seem to use most of the available blended resources. They put a lot of effort and work into the course.

The students achieve well, except for a small group of students who have challenges in other courses as well. Videos are highly motivating and useful, and students often request more videos. Students also request more ‘flipped classroom’ adaption and less plain PowerPoint lectures, even though students emphasize lectures as their most important learning element. This is probably due to many hands-on activities during the lessons. Both the individual and team-based work requirements seem to have been popular initiatives that inspire students to work.

**Suggested course improvements:**

- Many students say they would rather do more programming together with the professor, than listen to PowerPoint presentations where theory is explained. The role PowerPoints play in the course have already been reduced a lot, but even more of the theory explanation can and will be adapted to videos.
- Students say technical terms are difficult and confusing to students not familiar with these. Creating quizzes and Kahoots can possibly be useful for students to learn such terms, especially during the first half of the course.
Some students still report that the pace in lectures is too quick, but far less than before the video production. It is impossible to find a pace that suits everyone when a class consists of 80–90 students, but having more prepared students makes it easier for students to follow.

There has only been time enough to produce a limited amount of videos. The students have a very clear desire for more movies, so it could be useful to prioritize video production.

Due to time constraints, some of the videos were published after lectures, as rehearsal videos. More videos as preparation for lectures are requested.

It also seems important to prepare students for more self-driven activities than they normally are used to from prior education, to stimulate motivation and work effort.

REFERENCES


