

## Using Short Videos in an Introductory Programming Course

Kurt Nørmark

**Abstract:** Video resources are gradually being introduced in a first year programming course. The course is about imperative programming in C. After each of the first two years with video lectures the students have been asked about their assessments and opinions. The main parts of the paper are based on evidence from two questionnaires. It is concluded that a vast majority of the students find it very attractive to learn about imperative programming via videos (with or without supplementary coverage from a textbook). The paper describes the status of an ongoing activity. In addition to conclusions on past experience we also carry out a discussion of future plans for using video resources in the course.

**Key words:** Screen casts; Short videos vs. long recordings of lectures; Imperative programming course; Student opinions.

### INTRODUCTION

In this paper we will report about our experience with use of short videos in an introductory programming course. The reporting period includes two runs of the course. In the first run (in the fall 2012) a small number of 10-12 minutes videos were used in a sporadic way. In the last run (in the fall 2013) a larger number of videos were used in a more systematic way to cover selected topics in the course.

After the first run of the course the student's opinions about video support were collected in a brief questionnaire. Based on this input, the use of video in the second run was prepared and designed. A slightly extended version of the questionnaire was used after the second run of the course.

The remaining parts of the paper are structured as follows. In the first section following the introduction we describe relevant details of the course. In the next section we describe our general approach to use of video in the course. Next follows a section about the experiences collected from the first run of the course, where videos were used only sporadically. This is followed by a section where we discuss the experience from the second run of the course. The paper ends with discussions and conclusions.

### ABOUT THE COURSE

The programming course, from which experience is drawn in this paper, is called "Imperative Programming". The course is attended by first semester students of the bachelor programmes in computer science and software engineering at Aalborg University. In 2012 the initial number of students attending the course was 157. In 2013, 186 started on the course. The course is rated to 5 ECTS points.

The imperative programming course is based on the C programming language. The course is divided in 14 sessions, each lasting half a day. A course session includes a two hour lecture (in an auditorium) and a two hour exercise slot. The exercises are carried out in the student's group rooms, and they are supported by a number of teaching assistants.

As part of their preparations the students are expected to solve some programming exercises at home (on an individual basis). In addition, the students are expected to solve programming exercises in groups - in the exercise slot - just after the lecture. Finally, the students are offered a weekly programming assignment for submission (upload) - on an individual basis. The submitted programs are evaluated by the course teachers, and the students get individual, qualitative feedback [4].

The oral course exam is based on a larger (20 hours) programming assignment, the solution to which must be uploaded (in the same way as the weekly programming assignments mentioned above). During the exam the students are questioned about their programs, and some discussions are carried out.

Side by side with the programming course, the students follow a mathematics course (in linear algebra) and a course about problem based learning (PBL). Both of these courses are rated to 5 ECTS point as well. The remaining 15 ECTS points of the semester are devoted to project work, following the PBL approach used at Aalborg University [3]. The project work calls for design and implementation of a C programs as part of the problem solution.

### ABOUT THE VIDEOS

All videos for the imperative programming course are screen casts.<sup>8</sup> The videos developed for the imperative programming course fall within the following categories:

1. Videos about software installation and use of tools (5).
2. Videos about C operators (16).
3. Videos about control structures (2).
4. Videos about functions and parameters (3).
5. Videos associated with exercises and solutions to exercises (4).
6. Videos about terminology (5).

The numbers supplied in parentheses designate the number of videos in each category (as of 2013). The average length of the videos is 9 minutes and 30 seconds. The shortest is only 1 minute and 33 seconds, the longest 23 minutes and 50 seconds.

The videos in the first category are used early in the course to support the installation, setup, and initial use of compilers, editors, and web-based teaching materials. The videos in the second category represent the most complete group of videos. The ambition has been to cover the landscape of C operators rather systematically, together with areas of C programming which are natural to discuss with operators as the starting points. Category three and four - about control and functions - still suffer from lack of coverage.

The videos in category five have two different purposes: (1) To introduce an exercise (what is the problem), and (2) to show how to solve the exercise. Because some students have a hard time to understand the formulations of some exercises, video formulations and introductions may often be attractive alternatives to written formulations. Solution to exercises may involve a recording of a sample solution process, where the teacher shows a possible concrete way to program the solution (actually entering the program in real time, while talking about its qualities). Thus, the actual creation of the program is shown, along with the considerations that led to that particular program (on the audio track of the video).

The videos in the last category take for granted that the students understand substantial parts of imperative C programming curriculum. The goals of these videos are to rehearse the programming terminology. The students are especially motivated to watch such video as a preparation for the oral course exam - because mastering the terminology plays an important role in a successful oral exam.

### FIRST RUN EXPERIENCE

After the 2012 version of the course, the attitudes of the students regarding the use of video was examined through a web-based questionnaire. Each question was accompanied by a comment field that allowed for "free text" qualitative feedback. A total

---

<sup>8</sup> A screen cast consist of a video recording of a selected area of a computer screen accompanied by an audio narration.

of 59 students answered the questionnaire, and many students gave their opinions in free textual style. The questions and the quantitative results are shown in Appendix A of the full paper [5]. In the rest of this section we discuss the overall outcomes of the study, and we summarize the qualitative feedback provided by the students.

An overwhelming amount (53 out of 59 students) find that short, focused videos that complement the lectures will be useful (to a high degree or to some degree) for the imperative programming course. It should be noticed, however, that these answers are based on only a few examples of such videos from 2012. Thus, in this run of the course, the students have not yet experienced such “short and focused videos” on a larger scale.

Some student points out that video lectures are most desirable when they are related to “difficult” areas of the course. A student emphasizes that the video should be fast paced, and without “ritual” introductions. If not fast paced, some students will be tempted to do fast forwarding or simply find other means to obtain the knowledge. Another student ask for exercises (or other similar activities) in the slipstream of the videos.

It is interesting - and surprising - that 36 students (out of 59) state they prefer to watch a video, instead of reading a textbook when they are learning how to program. It is, however, clear from the comments attached to the questionnaire that the students also need “textual teaching materials”. A comment from one of the students probably “hits the nail”: *Books for theory, video for practical training*. One student is concerned with the time consumption involved by watching video, compared to reading a textbook. A good video structure - and use of video chapter markers - is called for.

Traditional university lectures have been around for centuries. By nature, most lectures convey information from the teacher to the students - with only little dialogue or interactivity. It is undoubtedly a concern of many university teachers (and students as well) if extended use of video lectures will make the traditional lectures obsolete. If traditional lectures are replaced by video lectures, university teaching loses one of its primary characteristics.

One student states the following: *I do not stay away [from the lectures]. I have learned extremely much from the lectures. The lectures are the live version of teaching*. Although technology may offer good alternatives, being present at the live event will probably always be more attractive than watching a streamed version of the event, let alone a rerun.

The majority of the students prefer short videos over full recordings of traditional lectures. However, one third of the respondents would wish to have both short videos and long videos (where “long videos” are recordings of the lecture in an auditorium). This represents - in fact - a remarkable demand for *full lectures on video*. One student states that he often misses details during a lecture (because of taking notes etc.), and that it would be attractive to recapitulate parts of the lecture via video. Another student proclaims that he simply does not understand why live streaming - or a least recording - of lectures are not more widespread. If - in the future - a lecture can be recorded by “a simple push on a button” it seems hard to reject the student’s desire for *full lectures on video*.

The students reflect a broad range of applicability of short course videos. There is substantial interest in using the videos both before the lectures, after the lectures, in exercise sessions, and as preparation for the course exam.

Less than a third of the respondents watched at least one course-relevant video from the internet on a weekly basis. This was a surprise, not least in relation to the large volume of existing videos about programming on YouTube and similar sites. There seems to be several possible explanations. One student states that he does not necessarily trust videos from the internet, unless they have been recommended or endorsed for the course. Another student states that *they [course-related videos on the internet] are*

frequently of very poor quality. Some university courses are - in the eyes the students - difficult to grasp;

Some students talk about *complex courses*. In such courses some students have a very hard time reading and understanding the course textbooks. A student writes that *it would be fantastic [if video lectures] were available in complex courses like Computer Architecture and Discrete Mathematics*.

	Number of videos in category	Average number of plays per video	Average total play time (minutes) per video
Software installation and use of tools	5	157	337
C operators	16	92	339
Control structures	2	89	261
Functions and parameters	3	77	424
Exercises and solutions to exercises	4	60	321
Terminology	5	283	996

Figure 1: A tabular presentation of the video categories, the number of videos in each category, the number of times each video (in the category) has been played (in average), and the number of minutes each video has been played in average. All data in the table are collected from September 1, 2013 to February 1, 2014.

## SECOND RUN EXPERIENCE

Almost all the trends, as reported in the previous section, reappear in the data collected in the fall of 2013. The quantitative data from 2013 can be seen in Appendix B of the full paper [5].

Whereas the first run of the course (as reported in the previous section) was based on very limited concrete experience with use of video lectures, the 2013 run of the course is based on much larger body of short videos (35 videos in total). In the 2013 run of the course it is evident that almost all of the students have made use of the videos.

In 2013 we created five short videos on imperative programming terminology. In general, the terminology videos rehearse the course terminology while showcasing a number of sample C programs. More specifically, in these videos the names of the programming concepts are articulated and emphasized with large textual overlays during the course of a video. As it appears from Figure 1, the videos about terminology turned out to be the most popular. As can be seen in the Figure, each of the five video was in average played 283 times, and each of the five videos were played for a total of 996 minutes (in average). Each time one of the terminology videos were shown, they were played in between 63% and 80% of their total playing time. (Videos in the other categories typically only played half of their total playing time, meaning “a typical student” stopped watching a video after having played half of it).

As it clearly appears, the students appreciate short videos side by side with the usual and traditional lectures in an auditorium. We also asked if the students would like to make the transition to a *flipped classroom* setup [6] (where the traditional lectures are supported by video, and where time spent at university is devoted to exercises and similar activities that call for active participation). Only 6 out of 55 students (11%) like this idea. Almost half of student (45%) answers “no” to this question.

## DISCUSSION

The goal of this project has been to create short videos that help our students understand the central topics in imperative programming. For that reason we have chosen to produce the video in the native language of the students (Danish). We are convinced that English versions of the videos would have decreased the outcome among the students. For the same reason, the videos discussed in this paper can never be part of a MOOC (Massive Open Online Course) setup [2, 1]. The amount of potential students, who are interested in C programming, and who understand Danish, will never sum up to a “massive amount”.

A high audio quality has also been emphasized during production of the videos. It is easier to get a good audio quality in short videos (produced in a “studio”) compared with a situation where the video (and sound) is recorded in an auditorium. One of the students comments on this as follows: *In general, the sound and the clarity of short videos is much to prefer [compared to long videos recorded in an auditorium].*

When asked about their preferences - textbook vs. video - some students emphasize the value of being able to switch between *learning modes*. One student says: *When you have lost your energy reading in the textbook, you can usually find some energy to watch a video about the topic, a hereby learn something nevertheless.* Another student bring in the concept of visual learners: *I clearly prefer both a textbook and a video, because the video is really good for those of us who are visual learners.*

We have disseminated all videos on YouTube. This frees the author and the university to provide a streaming video server. As long as we can avoid commercial advertisements in the videos, we are happy with the dissemination through YouTube. We have also benefited from the analytic tools provided by YouTube (for the analyses shown in Figure 1, and beyond). The videos are hidden from the general YouTube viewers (in the sense that the videos can only be seen if a YouTube URL is known). We may consider to make some of the videos public in the future. However, videos that reveal solutions to exercises must be kept “secret” if the exercises should be reused in the next round of the course.

For the next round of the course (fall 2014) we consider to reorganize the course, and to reduce the time reserved for lectures in the auditoriums. More specifically, the auditorium lecture time should be reserved to provide overviews of the course topics. Selected details should be supplied in short video lectures which the students will be asked to watch as part of their preparation. This will free some time for other activities, such as programming “while at the university” and reiteration of selected topics “just in time” before the exercise sessions. This can be seen as a *half flipped classroom* approach. From the attitudes reflected by the students we would not dare to implement a full flip of the classroom.

## CONCLUSIONS

The most important conclusion from our studies is that the students react very positive to short video lectures in the imperative programming course. From our studies it may even be fair to state that many students demand such videos. It also seems to be the case that video resources are requested more broadly by the students - across our portfolio of courses.

Contrary to the apparent student interest we have observed from the number of video plays that most videos have not be seen by all students individually. (Only the videos about terminology seem to have attracted “an overwhelming amount of plays” relative to the number of students). In addition, we can see from the analyses that – in

average - a run of a video stops when half of it has been played (the terminology videos plays in average until 75% of it has been seen).<sup>9</sup>

From an analysis of the student's attitudes it appears that the topics of the videos should be considered carefully, relative to the curriculum of the course and relative to workload that we impose on the students. Some topics (such as "theory") may be best covered via textbooks. Other topics (such as "practical mastery" and understanding of dynamic processes) are good candidates for video exposition. The actual imperative programming steps, and the considerations behind these, seem very well-suited for video recording. Overall, many students request a mix of traditional textbook resources and short supplementary videos. In the future it may be attractive to provide teaching material that integrates both of the approaches into each other.

As a caveat, it must be observed that final conclusions are difficult to draw without evidence from potential higher passing rates or higher grades. It is, however, difficult, awkward, and perhaps even irresponsible to arrange for scientific test where one half of the students use textbooks, and the other half use video resources. We would definitively refrain from such a test in the scope of our imperative programming course.

It is evident from our studies that short videos are more popular than (one or two hour) long videos of traditional lectures. However, approximately one fifth of the students request both short videos and long video recording of the traditional lectures. If, in the future, it becomes easy<sup>10</sup> to record the lectures in the auditorium, it seems worthwhile to do so. Our study reveals, however, that we should be prepared to accept that 30-40% of the students will occasionally not participate in the traditional lectures. This will undoubtedly lead to controversies about "waste of resources" for preparation of lectures, and it may end up closing the traditional lectures all together - or perhaps more likely, closing the video recaps of the lectures.

## REFERENCES

[1] Manuela Aparicio, Fernando Bacao, and Tiago Oliveira. Mooc's business models: Turning black swans into gray swans. In Proceedings of the International Conference on Information Systems and Design of Communication, ISDOC '14, pages 45-49, New York, NY, USA, 2014. ACM.

[2] John Daniel. Making sense of MOOCs. Journal of Interactive Media in Education, 2012.

[3] Finn Kjersdam and Stig Enemark. The Aalborg experiment - project innovation in university education. Aalborg University Press, Niels Jernesvej, DK-9220 Aalborg, Denmark, 1994.

<http://www.adm.aau.dk/- rektor/aalborgekperiment/engelsk/preface.html>.

[4] Kurt Nørmark. A web support system for submission and handling of programming assignments. In The proceedings of E-Learning'11 - E-Learning and the Knowledge Society, August 2011. <http://people.cs.aau.dk/~normark/programming-assignments-paper.pdf>.

[5] Kurt Nørmark. Using Short Videos in an Introductory Programming course. Full version with appendices. July 2014. <http://people.cs.aau.dk/~normark/video-paper-full.pdf>.

[6] Rebecca H. Rutherford and James K. Rutherford. Flipping the classroom: Is it for you? In Proceedings of the 14th Annual ACM SIGITE Conference on Information Technology Education, SIGITE '13, pages 19-22, New York, NY, USA, 2013. ACM

---

<sup>9</sup> It is very difficult to conclude strictly about playing times and "coverage". Some viewers may stop watching immediately, for instance just checking that the video is in fact available. Others may run the video from one end to the other without really paying attention to it.

<sup>10</sup> In this context, "easy" means "to push a button" in the AV panel.

**ABOUT THE AUTHOR**

Associate Professor Kurt Nørmark, PhD, Department of Computer Systems, Aalborg University, Denmark. Phone: +45 9940 8896. E-mail: [normark@cs.aau.dk](mailto:normark@cs.aau.dk).

**The paper has been reviewed.**