https://doi.org/10.36315/2021end104
Education and New Developments 2021

TEACHERS EXPERIENCE WITH INTRODUCING PROGRAMMING IN DIFFERENT COURSES FOR NON-COMPUTER SCIENCE STUDENTS

Martyna K. Fojcik[1], & Marcin Fojcik[2]
[1] *Faculty of Humanities and Education, Volda University College (Norway)*
[2] *Department of Electrical Engineering, Western Norwegian University of Applied Sciences (Norway)*

Abstract

Digital literacy has become more and more important in the last decade, and many people predict that in the future, the need for digital skills will be even more crucial than it is today. The dynamic development and use of technology are becoming increasingly common in all areas of life, changing demands of modern life and the labor market, which makes it necessary to educate students from many different study-programs on how to use different digital tools and how to program.
Depending on different professions, there are different requirements on what it means to have digital literacy. For some it is most important to know how technologies are created or to use the product effectively, for others it is the security of data transfer that is essential. The different professions have different needs for digital literacy and different use for programming skills.
Teaching computer programming can be particularly difficult in the case of introducing programming for non-computer scientists. While computer science itself (programming) is relatively well described in the subject's literature, the use of programming in other professions is not well defined. There are different suggestions, recommendations according to the level of education (primary, secondary, higher) or the study-programs the students take. There is no definition of what digital literacy is in different professions, what it means to know computer programming in different professions, and to what extend the students from non-computer science courses should master digital literacy and programming. That can cause challenges for the teachers and students in non-computer science professions that are required to know computer programming for their future jobs. There is no doubt that academic computer science skills for non-programmers can mean/contain different knowledge depending on course curriculum, teachers' experience, chosen literature, but the level of obtaining digital skills should be comparable, adequate, and relevant for the modern citizen.
This article presents requirements, some descriptions/cases of introduction to programming for non-computer scientists from a teacher's perspective. An adaptation of the general programming knowledge into the specific need of different subjects. The data is collected from higher education teachers that have different backgrounds and are teaching at different study-programs to get various views and experiences. The analysis of the findings uses SOLO-taxonomy to compare to what extend the different courses introduce programming to students.

Keywords: *Digital literacy, programming for non-computer scientists, introduction to programming.*

1. Introduction

According to research, almost two-thirds of all new jobs created since 2010 in America require either high or medium ICT skills (World Economic Forum, 2020). This trend is similar in other countries around the world and forced by the rapidly changing technology market. The dynamic development of information technologies and services and the changing demands of modern life are leading to changes in the labor market and, consequently, in all levels of education. To fill this gap and supply the needed digital competencies for society programming has been introduced in many non-computer science professions.

Today, university students have many subjects related to information technology, programming, or the use and distribution of digital machines, but many subjects focus on theoretical knowledge rather than skills development, or develop skills in computer science programming rather than focus on what kind of programming skills are needed in different professions. On the one hand, there is computer science and programming as a part of computer science, and on the other hand - teaching programming people from other fields, not as a part of computer science. This means that many students learn programming skills from the creators' point of view, even though they are studying the use of programming in their professions

rather than creating a computer program. A key challenge for today's universities is to identify which programming skills are being taught and how to teach them according to the students' study program. What are the requirements for programming in each profession? How can this be done to convey an important element and at the same time not overload the curriculum with elements that are unnecessary for this course? (Does a math teacher in primary school need to learn how to sort binary trees?)

This article will present some challenges with teaching digital literacy, and some definitions on digital literacy/digital competencies presented in EU and national documents. Then it will present two courses, one automation engineering course from Western Norway University of Applied Sciences (HVL), and one teacher education course from Volda University College (HVO). These two courses are introducing programming for non-computer scientists to, amongst other things, support students' development of digital literacy through programming. These courses represent only a part of students' skill development in their study programs, but they were chosen for this comparison since they are mainly focusing on the central part of digital literacy in these study programs. The chosen courses were analyzed according to the SOLO-taxonomy and then compared to the EU and national requirements on digital competencies.

2. Challenge in teaching digital literacy

Regardless of the field of knowledge, new competencies related to digital literacy and their practical application are needed both in professional development and private lives. These competencies depend on the skills of the workers, which in turn are based on knowledge, experience, and additional (non-professional) skills. Digital literacy is a broad gathering of different theoretical and practical skills and competencies on how to use technology and different functions that comes within, how to be aware and safe using technology, how are digital media and digital systems constructed, and how to use it in an efficient, secure and functional way (Buckingham, 2010). These skills can be divided into soft and hard competencies. Hard competencies are developed through academic education, work experience, and technical knowledge, while soft skills can be understood as a set of personal, interpersonal, and managerial competencies, like time-management, professionalism, reliability, ability to work under pressure, communication skills, creativity, and self-confidence. Andrews and Higson (2008) noticed a difference between graduates' skills and employers' expectations, where the set of skills the students develop during education weren't enough for the employers. The future employer needs the student to have not only soft and hard skills fitted for the work but also experience and practical knowledge about how to use technology. The Gallup-Purdue indicator provides statistics on the employment of graduates (Gallup, 2014), and according to their statistics students who have worked on projects lasting at least one semester or who have had a job or practice in the same area as their studies double their chances of getting involved in work later in life because they had more experience in developing practical skills needed in their professions.

The challenge in higher education is to adapt to the form of knowledge transfer to help students develop these practical skills during the course. To support skill development needed in each profession and prepare students for future employment. Today's students expect new forms of teaching that will help them to acquire practical skills, especially such as teamwork, cooperation, and responsibility, but the structure of teaching in higher education is not prepared for such forms. A higher education teacher might explain theoretical perspective and show technological tools and functions for the student, but that is not enough to let the student develop his own skills. The challenge for today's teachers is to find out what existing and new approaches are needed in teaching and how best to meet students' needs to facilitate students' development of digital literacy.

3. EU and national requirements

The Council of the European Union, first in 2006 and then in a new Recommendation of 2018, has placed digital competence as one of the eight key competencies for lifelong learning. The Recommendation defines digital competence as self-confidence, critical thinking, and responsibility for using digital technologies and describes it in terms of knowledge, skills, and attitudes (Council Recommendation, 2018). International documents provide guidance for improving and understanding digital competence, highlighting the need to plan innovative educational activities and assess student competence levels.

The need for developing digital competencies have been researched in project DigComp 2.0 (Vuorikari et al, 2016), which identifies the key components of digital competence in 5 areas which can be summarised as follows:

- **Information and data literacy**: To articulate information needs, to locate and retrieve digital data, information and content. To judge the relevance of the source and its content. To store, manage, and organize digital data, information and content.
- **Communication and collaboration**: To interact, communicate and collaborate through digital technologies while being aware of cultural and generational diversity. To participate in society through public and private digital services and participatory citizenship. To manage one's digital identity and reputation.
- **Digital content creation**: To create and edit digital content. To improve and integrate information and content into an existing body of knowledge while understanding how copyright and licenses are to be applied. To know how to give understandable instructions for a computer system.
- **Safety**: To protect devices, content, personal data, and privacy in digital environments. To protect physical and psychological health, and to be aware of digital technologies for social well-being and social inclusion. To be aware of the environmental impact of digital technologies and their use.
- **Problem-solving**: To identify needs and problems, and to resolve conceptual problems and problem situations in digital environments. To use digital tools to innovate processes and products. To keep up-to-date with the digital evolution.

4. Requirements for non-computer science professions; engineers and teachers

Norwegian National Curriculum Regulations for Engineering Education (UHR-MNT, 2020) describes what students should accomplish through developing digital competencies during a course:

- The candidate has knowledge of software useful at work and broad engineering skills, including basic programming skills.
- The candidate may work inappropriate physical and digital laboratories and master methods and tools as a basis for targeted and innovative work.
- Candidates will be able to identify the security, vulnerability, privacy, and data security aspects of ICT products and systems. Based on these descriptions of learning outcomes are learning outcomes in ICT programming and security

National curricula also describe the requirements for programming skills - and not only for programmers. Digital literacy plays an important role in the development of future society and industrial solutions. This means that programming, software, and computer technology are taken into account as technologies for the development and implementation of many system solutions. Because of this development, the engineering education framework requires basic programming skills in all fields of engineering.

In a similar document presenting teacher education should develop (NRLU, 2016), it is written that basic skills: reading, counting, writing, speaking, and using digital technology are both a prerequisite for developing academic knowledge and part of academic competence in all subjects. Teacher education supports the development of students' basic skills at the subject level. Teaching practice includes the teaching and use of digital resources. A teacher should have knowledge of the use of various teaching aids, both digital and other, and of the possibilities and limitations of these aids.

5. SOLO-taxonomy of introduction to programming at universities

In a previous study, the authors analyzed two introductions to programming courses (Fojcik et al. 2020) and compared the content of the courses based on SOLO-taxonomy (Structure of the Observed Learning Outcome) by Briggs and Collis (1982). This taxonomy is used to evaluate the observed students learning outcomes in a lecture or a course. Based on the analysis of Fojcik et al. (2020), some criteria for developing programming skills were described into an extended model of 5 phases of SOLO-taxonomy. The overall results and description of skills developed in each programming course are presented in Table 1.

Table 1. Compared of HVL and HVO courses based on Fojcik et al. 2020.

Level	Name	Programming competence (computer science)	Programming for teachers (HVO)	Programming for auto. engineer (HVL)
5	Extended Abstract	Develop the skill of creating, testing, and operating programs, making and distributing classes, libraries for multiple uses, cooperation, and collaboration with other programs and users.	Show pedagogical, didactical, and content knowledge of programming, the skill to use simple programs, understanding and explaining computational structures, supporting computational thinking in students/pupils in all levels of education, etc.	Model of software as a model of real action/automation, libraries, interfaces, analogies between real-world and software
4	Relational	Create a working program, divided into different elements (class, structure, etc.), full debugging, collaboration and cooperation with the user, saving state, file handling, network, database, etc.	Explain why the program works, reading and understanding other's code to debug programs, etc.	Dividing action into smaller, repeatable elements, reusability, safety, friendliness of the software
3	Multi-structural	Show sophisticated elements in coding, combining several different actions, compiling user interface, etc.	Independent writing of the algorithms and create simple working programs, explaining efficiency and accuracy of their own work, etc.	Repetitions of actions, more complicated elements, user interface, error-handling
2	Uni-structural	Show simple skills, using variables, comparing elements, simple loops. Everything in one file, with no dividing in sections, classes or structures, etc.	Independent writing of simple commands/sequences, using simple loops and/or if-sentences, etc.	Understanding the idea of planning and operations order, the definition of variables (connected with real measurements), and simple operations on them
1	Pre-structural	Barely anything or just a very general or divided knowledge – wrong assumptions and unconnected information, etc.	Barely anything or just a very general or divided knowledge – wrong assumptions and unconnected information, etc.	Barely anything or just a very general or divided knowledge – wrong assumptions and unconnected information, etc.

6. Comparison and discussion

Depending on different needs, there is no coherent plan for teaching digital competencies. Requirements from the employers and the European Union indicate basic digital literacy as the creation, modification, use of digital components understandably and responsibly. Many of these elements are reduced in university programs. The role of universities is to train professionals, but digital skills are often described as a basic level, and usually more or less combined with programming skills, even though they might not intersect in EU and national requirements for digital competencies.

There is a gap between requirements at different courses and overall requirements at higher education levels when teaching digital literacy. This is most evident in teaching programming. Programming can be presented as the most important digital skill for the students, but not all of the students should be programmers. That is not their goal when they choose to study engineering, teacher education, biology, or any other professions, but all students must learn to use digital tools responsibly and safely both in their professional and private life.

Another challenge is how to teach non-computer scientists to program. The results of SOLO-taxonomy comparison of HVL and HVO courses show that the development of programming skills should be customized to the pedagogical and didactical requirements of the profession they are thought in, and not necessarily as computer scientists.

The analysis based on SOLO-levels of the course content shows differences in what students learn about programming and what they do on assignments to what they perform on exams and evaluations (Fojcik et al, 2020). This result shows that students understand the theory of programming and using technologies at a higher level than they can perform on practical exercises in using programming or technology. The challenge in higher education is to support students to develop understanding and practical skills to match their theoretical knowledge on the same topics.

References

Andrews, J., & Higson, H. (2008). Graduate employability, 'soft skills' versus 'hard' business knowledge: A European study. *Higher education in Europe*, *33*(4), 411-422.

Buckingham, D. (2010). Defining digital literacy. In *Medienbildung in neuen Kulturräumen* (pp. 59-71). VS Verlag für Sozialwissenschaften.

Council Recommendation (2018). On key competences for lifelong learning. In Official Journal of th European Union *189, 4.6.2018, p. 1–13. Retrived form* https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv%3AOJ.C_.2018.189.01.0001.01.ENG

Fojcik, M. K., Fojcik, M., Sande, O., Refvik, K. A., Frantsen, T., & Bye, H. (2020). A content analysis of SOLO-levels in different computer programming courses in higher education. In *Norsk IKT-konferanse for forskning og utdanning* (No. 4).

Gallup (2014), *Great Jobs Great Lives. The 2014 Gallup-Purdue Index Report*. Retrieved from https://www.gallup.com/services/176768/2014-gallup-purdue-index-report.aspx

NRLU (National Council for Teacher Education) (2016). *National Guidelines for the Primary and Lower Secondary Teacher Education Programme for Years 5-10.* Retrieved from: https://www.uhr.no/_f/p1/iecd98eeb-d012-44ce-b364-c8787ca51a95/national_guidelines_for_the_primary_and_lower_secondary_teacher_education_programme_for_years_5_10.pdf

Recommendation of the European Parliament and of the Council (2006). *On key competences for lifelong learning*. In Official Journal of the European Union 3*94, 30.12.2006, p. 10–18. Retrieved from* https://eur-lex.europa.eu/eli/reco/2006/962/oj

UHR-MNT (2020). *National Curriculum Regulations For Engineering Education*. Retrieved from https://www.uhr.no/_f/p1/i0cb3d399-4d21-4317-8978-e4f44c2306c1/nasjonale-retningslinjer-for-ingeniorutdanning-vedtatt-av-uhr-mnt-november-2020.pdf

Vuorikari, R., Punie, Y., Carretero, S. & Van den Brande, L. (2016). DigComp 2.0: The Digital Competence Framework for Citizens. Update Phase 1: the Conceptual Reference Model. Retrieved from https://ec.europa.eu/jrc/en/publication/eur-scientific-and-technical-research-reports/digcomp-20-digital-competence-framework-citizens-update-phase-1-conceptual-reference-model

World Economic Forum (2020). *The Future of Jobs Report 2020*. Retrieved from http://www3.weforum.org/docs/WEF_Future_of_Jobs_2020.pdf