Flipped Classroom Real-World Activities for Learning Open Computing Concepts

https://doi.org/10.3991/ijet.v17i07.25767

Ivana Bosnić[1(✉)], Igor Čavrak[1], Bastiaan van Loenen[2]
[1] FER, University of Zagreb, Zagreb, Croatia
[2] Delft University of Technology, Delft, The Netherlands
ivana.bosnic@fer.hr

Abstract—Various "open" concepts in Computing, such as open standards, open data, open licenses or system interoperability are becoming more important in the professional lives of software engineers. However, students usually do not receive a systematic education about these concepts; rather they sporadically learn about a subset of these topics. This paper presents the revised version of the *Open Computing* course in the University of Zagreb, Faculty of Electrical Engineering and Computing (FER), which teaches a clear set of current topics focused on open data and correlating concepts. The new e-learning course is carried out using the "flipped classroom" educational method; students construct their knowledge in a set of real-world mini-activities throughout the course, instead of passively learning from the official course resources. In this paper, we discuss our flipped classroom activities, their relation to revised Bloom's taxonomy of educational objectives, and present the evaluation of the first course instance using this model. Students' feedback shows they welcome this change in approach, finding the course useful and interesting, and preferring this method to the traditional full lecture setting.

Keywords—open data, open computing, education, e-learning, flipped classroom

1 Introduction

The need to teach students open computing concepts developed almost two decades ago, when the popularity of open systems started to rise, primarily thanks to the development of the Internet and World Wide Web. Issues of those times, such as interoperability, open formats, standards and security, also exist now, with a different focus and emphasis, and including new concepts such as open data, open licensing, open source communities, and open science. This knowledge is needed and valued, especially when moving on from universities to the world of companies, interoperable products, data exchange, huge IT systems or government projects. However, not many courses focus entirely on open concepts. Education on these topics sometimes plays a small part in more extensive courses on other topics, where a section of open concepts is used as a basis – or a tool – to achieve some other course-specific purpose.

The *Open Computing* course started almost 20 years ago, to fill the gap on the topics mentioned above, giving students an overview and basis of open concepts of those times marked with "browser wars"[1], fights or monopolies on file formats, character encoding mess and other issues that an average IT student of today doesn't even know about. The course was fully revised, redesigned and updated in the academic year 2020/2021, along with our new Faculty study program of Computing. In the newly designed course, we would like to bring the attention of our students to the open environment as a whole. This would include open data, open standards and formats, open systems, open licenses, open source projects, particularly interoperability and reuse, with all issues that these efforts bring, especially if being as open as possible. As this course is aimed at higher education students of Computing and related sub-fields such as Software Engineering, Computer Engineering, Computer Science and Information Technology, the course program and its activities is tailored for the technical audience. However, several topics at least partially include the interdisciplinary approach, so the students could get a broader image of the open paradigm wherever used. Moreover, these topics could be used – with some adaptations – even in non-technical educational contexts.

In addition to the content we also changed the form of the education. Instead of traditional lectures, we decided to implement the flipped classroom model, to better engage students in the course content and connect it better to challenges of this field. This course redesign aims to include several activities that would connect the educational content to the real-world context. While this is not a „real-world project-based course, "where students play roles of future employees or experience the team dynamics, the activities presented in this paper give a taste of real-world issues, which should be relevant when starting to work in a complex, data-oriented, interoperability-enabled setting. In this paper we present the content and new educational concepts implemented in the new course on *Open Computing* in the University of Zagreb, and present and discuss the student evaluation of the course.

The paper is structured as follows: after presenting the related work in Section 2, Section 3 presents the course in general, its topics and structure. Section 4 describes the flipped classroom approach we decided to use in the new course program, along with 11 course activities, each adding up to the course objectives. In Section 5, students' feedback and evaluation of these activities and the course are presented. Section 6 discusses the general lessons learned, in addition to our comments discussed when describing the activities. Finally, we conclude the paper with several proposals for future improvements in Section 7.

2 Related work

The related work for this paper concerns three topics: open computing, open computing education, and the flipped classroom concept. We discuss each of these briefly.

[1] Browser wars – a period of competition between several browsers on the market, at the beginning of a wide World Wide Web usage

2.1 Open computing

Open computing refers to open computer systems that can be accessed, shared and reused by anyone for any purpose. Open computing is characterized by the incorporation of open software standards, open formats, open data, open licensing, and open source coding. This concept is a key for the interoperability of systems and the fundamental to open science approach [1], [2], [3].

2.2 Open computing education

The concept of "open", comprising a broad set of topics is not often found in formal education. Open data, which is a substantial part of our course, is slowly finding its place in courses, not necessarily being a primary teaching goal, but a mere tool or resources for other educational purposes. In a study from 2020, 10 interviews with educators who use open data were presented. Out of these, 5 courses were in higher education [4]. Linked data, whether open or not, can also be used in education for different purposes, as investigated in [5].

However, to the best of our knowledge, there are no courses comprising more of the open topics to give students, especially students of computing, a broader basis for open systems as a whole. A recent study into the content of higher education programs in spatial data infrastructures confirms that most programs include some of the open concepts, but none has embedded the full open computing concept [6].

2.3 Flipped classroom

Flipped classroom is a teaching strategy where students prepare for the class in advance using some provided educational resources, leaving the time of in-class meetings for some type of active problem solving [7]. It advocates the use of class time for "guide on the side" instead of "sage on the stage" [8] and it might improve students' motivation and manage cognitive load [9], [10]. Along with the similar strategy called *peer instruction* [11], this method gained momentum in 21st century, especially with the development of Internet, allowing students to access videos and other rich content at home, often using some kind of learning management system [12].

While the method is most often used in primary and secondary schools, it is becoming important for higher education as well, in spite of universities having their traditional lecturing methods. Flipped classroom can also be successfully used in Computing courses at the universities. A survey on 32 papers related to programming courses is presented in [13], while a newer review on 20 studies reports on various course experiments, with the majority of CS1 (Computer Science 1) courses [14]. Both papers show that this strategy has a future in successful teaching computing courses. A set of flipped classroom strategies for teaching Computer Science is discussed in [15].

A model used for assessing how our new course strategies fit into the course is well-known, a widespread revision of Bloom's taxonomy of educational objectives [16]. Put in practice, a defined table template is used for placing course objectives and activities in two dimensions. The first dimension, *Knowledge* that learners should obtain or construct, is classified into four major types, ranging from concrete to abstract knowledge:

factual, conceptual, procedural and *metacognitive* grouping 11 knowledge subtypes. The other dimension, *Cognitive Process*, presents six types of thinking skills, ranging from lower to higher order: *remember, understand, apply, analyze, evaluate, create*. These two dimensions enable a classification of learning objectives and activities needed to achieve the objective. In this paper, we will use this model to present the activities and objectives they are planned to fulfill.

3 Open computing course at a glance

Open Computing is an elective course for students of Computing at the University of Zagreb, Faculty of Electrical Engineering and Computing (FER). It is currently offered as a joint undergraduate and graduate course. The number of students in the course per year is 20-50. The course is awarded 5 ECTS points, which corresponds to ~150 hours of students' time. The course topics and structure are presented in more detail in the following subsections. The lectures and other educational resources (in Croatian) are publicly available on the course webpage[2].

3.1 Course topics

The course tackles several topics – at the first glance not very related to each other. However, all these topics form a whole by introducing students to underlying concepts of anything "open" – including open data, open formats, open systems, open licenses, security and, most importantly: interoperability as a basis for any data/content reuse or communication among applications and systems. Therefore, our course topics are:

- Introduction to Open computing: history, examples, policies, myths and legends
- Introduction to standards, open standards
- Openness of data formats – binary formats, text formats, code pages
  - Data formats – CSV, XML, JSON
- Open data – definitions, principles, influence, interdisciplinarity, advocacy
  - Creating open datasets, formats and standards, open licensing, tools, portals
  - Metadata, vocabularies and vocabulary formats, linked data
  - Open data quality, evaluation models and frameworks
- Free and open – introduction to free software and open source, licenses, Creative Commons licensing framework, open source projects
- Distributed systems – introduction, models, types of communication, exchange of data, XDR – eXternal Data Representation, remote calls
  - Web APIs - Application Programming Interfaces, REST architecture and APIs, maturity levels, OpenAPI
  - Security in open systems – authentication, authorization, TLS/SSL, HTTPS, OpenSSL, OAuth, OpenID, OpenIDConnect protocols and tokens (JWT)

[2] Open Computing course webpage: https://www.fer.unizg.hr/predmet/or

The course teaches the basics of open computing. In this way, students can get the broader image of an open system or environment, tackling all levels of infrastructure, and touching on the interdisciplinarity of open systems, often neglected in purely technical study programs. We try to reconcile these differences in topic types by varying course elements as presented below.

3.2 Course structure

The course lectures last for 13 weeks, divided into two periods of 7 and 6 weeks, with two weeks of mid-term exams in-between. As the course update coincided with the COVID-19 pandemic, we prepared the new course to be run in a fully online mode, with interactive online activities in a flipped classroom approach. We find this situation beneficial to our course, as many activities were more easily done online.

The course consists of three main teaching and learning elements:

- *Course lectures* – each week, a video summary of lectures is presented as a recording, along with learning material and a set of various links to check out. Students should study these resources in advance. At the end of the week, we have a joint online meeting – an interactive session, varying in type from week to week.
- *Student activities* – in most weeks, students get one of flipped classroom activities (described in this paper), which is not obligatory but is worth ~2% of the course grade. This activity can be either individual or done in a group. It can also be synchronous – to be carried out during our joint online meeting, or it can be done at some other convenient time. In the latter case, a discussion will take place on the asynchronously submitted activity results during the next weekly meeting.
- *Laboratory exercises* – although we aim for an interdisciplinary approach and broader view, *Open computing* is still a technology-oriented course, so our students should get the practical know-how on the technical topics. Therefore, in addition to "small" student activities, our course has five laboratory exercises. In general, students are asked to develop a web application for a topic of their choice that includes open data handling – modeling, creating, licensing and providing an open dataset. They should create a web interface, describe the dataset metadata, specify and develop a high-maturity RESTful API for exposing the dataset and integrate it with another linked open dataset. We also include an exercise on getting involved in an open source project of student's choice.

Students are generally used to such "technical" laboratory exercises as a part of courses at FER. However, they are not used to small weekly activities, especially if these are not purely technical tasks, but include other views of the real-world examples. This is precisely what we are trying to achieve with the flipped classroom course concept presented in the next section.

4 Flipped classroom course concept and activities

4.1 Motivation

As the *Open computing* course focuses on real-world problems and issues that students would be exposed to in their professional work, we decided to implement the flipped classroom model not only by studying concepts in advance but also by solving small real-world activities, ranging from analyzing real-world examples to proposing complex solutions to real-world challenges. We provided both individual and group activities, but also at-home activities and those performed during our joint online meetings. We tried to provide a convenient amount of such tasks, obtain continuity throughout the semester, and not overburden the students, which would accomplish the opposite effect instead of getting them interested in the topics.

The following sections present an overview of the real-world activities, our motivation for including these, the level of learning objectives in revised Bloom's taxonomy that an activity aims for - both the knowledge dimension and the cognitive process dimension. We also included our comments after carrying out each activity. For more convenient reading, a list of activities, including their characteristics - Individual/Group activities (A/G), Asynchronous/Synchronous activities (A/S), is presented in Table 1.

Table 1. The list of flipped classroom activities (I= Individual, G= Group, A=Asynchronous, S= Synchronous)

Week#	Activity#	Activity name	I/G	A/S
1	1	Examples of open and closed systems	I	A
1	1g	Myths and legends of open computing	G	S
2	2	Open and closed file formats	I	A
2	3	Research on practical features of code pages (encoding)	I	A
3	4	Your own errors on manual data creation	I	A
4	2g	Real-world open datasets	G	S
7	5	Examples of Creative Commons content	I	A
7	6	Issue tracking in open source projects	I	A
9	7	External data representation formats	I	A+S
11	8	Evaluate an API!	I	A+S
12	9	Speed, but caution (data integrity and privacy)	I	S

4.2 Individual real-world activities

Activity 1 - Week 1: Examples of open and closed systems

Topic: Introduction to open systems.

Our motivation: Making students more attentive to concepts present in the real-world, which adhere to open/closed systems principles.

Bloom's taxonomy: B.C. Conceptual – knowledge of theories, models and structures; 2.2. Understand – Exemplifying.

Overview: Students were asked to provide two real-world examples of openness or "closedness" (or something "in-between"). It did not matter whether the examples were related to technology or not. Students had to provide the description of examples, categorize them and explain their level of openness. The answers were put in *Moodle Database* activity.

Our comment: While most of the examples were from the software engineering category (which we expected, as students were just starting with the course and are not used to critically observe the world in the level of openness), it was interesting to see some unexpected non-software examples, such as the dimensions of transport containers, printer ink cartridges, traffic lights, vehicle motors, music notation, light bulbs or smartwatches. We plan to continue extending this database of examples over the years, using it for discussion among students on their peers' observations.

Activity 2 - Week 2: Open and closed file formats

Topic: Openness of data file formats.

Our motivation: Motivate students to analyze the history and context of particular file formats, which differ not only in technical characteristics but also in the level of openness, legal rights and other properties.

Bloom's taxonomy: C.B. Procedural – knowledge of subject-specific techniques and methods; 4.3. Analyze – Attributing.

Overview: Each student was given two data types – file formats, intended for various purposes (text, document, image, audio, video). They had to research and shortly describe the format in terms of openness. Hint questions were provided, such as "*What is the history of format development? How is it licensed? Are there patents involved? Is it published as a standard? Is it widely used? Who supports it? Are there open/closed alternative formats for the same purpose?*" The answers were put in Moodle Database activity. Each file format was analyzed by more than one student, thus providing insight for discussion on answers during our joint online session.

Our comment: As students were mainly using the most popular pages for their research (such as Wikipedia, containing well-structured information on each file type), most of the facts in the answers were similar. Sometimes the students differed in their conclusions and discussions (for instance, one student would conclude that GIF is an open format, while the other would describe it as closed). Such examples of different opinions showed students how complex the discussion on openness could be.

Activity 3 - Week 2: Research on practical features of code pages (encoding)

Topic: Openness of data file formats.

Our motivation: Move from theory to real-world issues regarding code pages and character encoding, showing how these concepts influence both their programming experience and user experience when using the applications.

Bloom's taxonomy: C.B. Procedural – knowledge of subject-specific techniques and methods; 5.1. Evaluate – Checking.

Overview: Students were given the choice of several real-world topics related to character encoding. They had to investigate how theoretical background applies to these practical issues, explaining the reasons behind such behavior. Topic examples include:

- What happens in an SMS message after a Croatian diacritic character (i.e., č, ć, š, đ, ž) is inserted? How does this influence the length limit?
- How does the size and byte-content of a .txt document change related to the contained characters and the encoding chosen?
- Investigate the support for Unicode, collation and sorting of Croatian diacritic characters, including the digraphs (i.e., “nj”, “lj”). What is the length of such strings, calculated by string length functions in various programming languages?
- Try several examples of importing CSV documents into spreadsheets, depending on the document encoding and program options. Comment on failed imports.
- Research the level of *emoji* support in applications (word processors, browsers, mobile apps). Experiment with combined graphemes (e.g., skin color types of *emoji personas*). What happens if such *emojis* are not fully supported in a tool?

Our comment: This seemed like an interesting activity for students, providing a real connection between theoretical background and user/developers experience. Students had to provide their answer as a forum reply post before they were able to read others’ insights on the same task. Again, they were able to see the different approaches to the same issue, even different conclusions. We consider a success that all presented issues had a similar number of answers, meaning that they were equally interesting and of similar weight.

Activity 4 - Week 3: Your own errors on manual data creation

Topic: Openness of data file formats.

Our motivation: Learn from your own errors. Errors in learning are often considered harmful. This activity was meant to provide a positive experience of error-making.

Bloom’s taxonomy: D.C. Metacognitive – self-knowledge; 6.1. Create – Generating.

Overview: Upon theoretical learning about data representations in XML and JSON formats, students should manually code the structure of a simple phone book, both in XML and JSON, having the same characteristics, without a conversion data loss. The data representations had to be checked by a validation tool. Students had to report on and describe at least three errors that they did while solving this task. As always, their answers were also available for others to read and learn from them.

Our comment: Having students admit and write about their errors was a potentially risky task. Our students are quite shy, especially when having to reply non-anonymously, not to mention talking about their own errors. Contrary to our fears, students accepted this activity well; they admitted making even some fundamental errors (which they could have left out due to shame, but did not), and they discussed some issues in data mapping they had to solve. It would be wise to think about implementing more “error making” in a learning process in general.

Activity 5 - Week 7: Examples of *Creative Commons* content

Topic: Freedom and openness.

Our motivation: Evaluate how the principles of reusing the openly licensed content work in real-world.

Bloom’s taxonomy: C.C. Procedural – knowledge of criteria for determining when to use appropriate procedures; 5.2. Evaluate – Critiquing.

Overview: Students are invited to find two examples of reusing the *Creative Commons* licensed resources – one example of good and one example of inadequate referencing. Compared to the usual and too easy "find the examples of CC-licensed content usage" activity, this task increases students' critical skills by evaluating *reuse*. As reuse is the cornerstone of CC licenses, this reuse must be done properly, which is often not the case.

Our comment: As expected, some students did not understand the activity and instead found examples of CC-licensed content, not of reuse. Some others found examples of illegal use of content that is not necessarily *Creative Commons*. Still, the majority understood the task well and described in detail the referencing issues. We believe this activity, albeit small, brings a high level of understanding open licensing issues.

Activity 6 - Week 7: Issue tracking in open source projects

Topic: Freedom and openness.

Our motivation: Motivate students to get involved in open source projects by introducing them to open source project communities.

Bloom's taxonomy: C.A. Procedural – knowledge of subject-specific skills and algorithms; 4.1. Analyze – Differentiating.

Overview: After the general introduction to open source projects and communities, students were asked to find an open source project of their choice (preferably a more complex one) to research and describe how submitting issues and bug reports works in that community. Some helpful questions we proposed were: *Which tools are used to submit such reports? What should such a report contain? How are the already submitted issues handled?*

Our comment: We believe this activity was an important, yet simple, introduction to involvement in open source projects. The gap between a regular user and an active user in the community is often vast, and average students do not even think about getting involved. This topic should be further extended in our study program.

Activity 7 - Week 9: External data representation formats

Topic: Distributed systems.

Our motivation: Allow students to individually research a theoretical subtopic, prepare a mini-lecture resource, and (possibly) present it.

Bloom's taxonomy: A.B. Factual – knowledge of specific details and elements; 4.2. Analyze – Organizing.

Overview: As a part of the lecture on distributed systems, in particular external data representations, each student was given one (out of 8) representation formats (e.g., SOAP, RMI, YAML) to research and describe. Students were given a set of question hints and a template for the content creation. As multiple students were describing the same format, during the joint meeting we selected the best-prepared content for each format, and the author presented it in a 5-minute presentation.

Our comment: The content taught in this activity was a bit vague for students, as it was quite new in their education. We should think about whether to extend this task by asking to provide an example of usage, which would give more meaning to the activity but would also make it more difficult. We believe it is important that students try to prepare the educational resources for others and possibly even try presenting/teaching.

Nevertheless, it is a matter of discussion should such lessons become a part of the official exam content or not.

Activity 8 - Week 11: Evaluate an API!

Topic: Web APIs, REST.

Our motivation: Provide students a controlled environment for critical thinking about real-world technical solutions used globally, evaluating those and proposing development changes that would improve these solutions, thus preparing students for critical thinking in their future work.

Bloom's taxonomy: C.C. Procedural – knowledge of criteria for determining when to use appropriate procedures; 5.2. Evaluate – Critiquing (stage 1).

D.A. Metacognitive – strategic knowledge; 6.2. Create – Planning (stage 2).

Overview: This is a 2-stage activity on evaluating and proposing improvements for globally used public APIs of popular services. We prepared a list of 15 APIs (from the ones used locally in Croatia to *Wikipedia*, *Last FM*, *Instagram*, or *Spotify*). Students received three APIs each and had 60 minutes to check the API documentation. Based on the API features, they had to evaluate which level of maturity (0-3, based on a well-known Richardson's maturity model [17]) a particular API has. In the joint meeting, we compared the resulting evaluation grades, and students discussed possible differences in their evaluation. Additionally, for the next week, students had to individually propose how they would improve a particular API to gain a higher maturity level.

Our comment: This was a quite successful task, as it combined several quality points: a modern, exciting and valuable topic which students liked, a straightforward theoretical background and guidelines for assessment by using the abovementioned model, an easily comparable answer (only one number, 0-3) but which should be elaborated in details, and a "what next" stage. We find this to be a good combination of theory and practice, and a beneficial activity for the future.

Activity 9 - Week 12: Speed, but caution (data integrity and privacy)

Topic: Security.

Our motivation: Provide students a real-world situation regarding the security and authenticity of messages, evaluating the authenticity of a message.

Bloom's taxonomy: D.C. Metacognitive – self-knowledge; 5.2. Evaluate – Critiquing.

Overview: After theoretical explanations of various security concepts, students were introduced to the *OpenSSL* tool that can be used to check the authenticity in various forms. They had to analyze six messages and decide on their level of trust in the authenticity of this message. Although this was an individual activity, five sets of example messages were prepared, so a discussion on results would take place.

Our comment: This was a well-thought activity adequate for a real-world situation, with a probable interesting discussion on the results. Unfortunately, due to a mistake in preparing the example message, the results were not relevant, and the activity could not be accomplished. On the other hand, while not expected, students were very motivated to try solving this and tried several methods before we realized there was a mistake. An added - unexpected - bonus was that students actually had to convince the teacher that there is something wrong (which they successfully managed to do), and this turned to be a very good teaching moment for them – even the teachers can make mistakes; it is OK to confront them, especially when having good arguments.

4.3 Group real-world activities

Activity 1g - Week 1: Myths and legends of open computing

Topic: Introduction to open computing.

Our motivation: Motivate students to engage in discussion on the possible myths and legends on open topics based on their current knowledge and bias.

Bloom's taxonomy: D.A. Metacognitive – strategic knowledge; 5.2. Evaluate – Critiquing.

Overview: At the end of the first lecture, we presented students with ten myths/legends of open computing – discussable statements. Without explaining them, we selected five of them, grouped students into five breakout rooms for a short 5-minute discussion, having one myth per group. Afterward, a student representative of each group had to present their conclusion shortly, thus starting a broader discussion.

Our comment: This was the first synchronous activity of the course, where students were "unexpectedly" divided into groups and had to think through these statements and actively participate. We got some comments that this was unexpected and "too interactive," but we would argue that this is a matter of being used to passive lectures where the only thing students do is – listen (hopefully). We wanted to change that, and the first lecture is an excellent time to start making a difference.

Activity 2g - Week 4: Real-world open datasets

Topic: Open data 1: properties, influence and creation.

Our motivation: Engage students in thinking, discussing and proposing the usage of open data to solve a particular, ongoing issue in society.

Bloom's taxonomy: D.B. Metacognitive – knowledge about cognitive tasks, including appropriate contextual and conditional knowledge; 6.1. Create – Generating.

Overview: After the first week of learning about open data, students were divided randomly into groups of three members. They had 60 minutes to propose the solution to an issue of their choice by addressing these five steps: *(i)* choosing the topic/problem, *(ii)* finding at least two different open datasets for this purpose, *(iii)* analyzing and comparing the selected datasets, *(iv)* problem-solving and interdisciplinarity of the approach, *(v)* adding value and decision making. A structured presentation template was provided to expedite their work. After presenting the solutions, we discussed the issues such as the feasibility of proposals and other possible problems.

Our comment: Obviously, having a 60-minute limit for solving such a task is a bottleneck, leading to only starting to work on a solution. We are aware of this fact, but the main objective was not to solve an issue, but to engage in thinking about how open data would be used in such solutions, who would be the stakeholders, which information would we need and how would we get it, what would be the added value of using the open data, and so on. It was also the first time students had to show their creativity and engage with using open data in a real-world setting, so this was again a new experience. Students found this activity quite helpful, stating that they finally understood what open data really is about and what it can be used for. More details and the evaluation of this task can be found in [18].

4.4 Revised Bloom's taxonomy of activity objectives

The mapping of our activities to dimensions of the revised Bloom's taxonomy of learning objectives is presented in Table 2. For clarity and brevity, the complete objectives of activities were omitted but could be extrapolated from "our motivation" sections. As explained in [16], activities adding up to an objective could also be presented in these kinds of tables. The activities contain their number, and the *type.subtype* label for each of two dimensions. Please refer to the activity descriptions above for the complete type names.

Table 2. Revised Bloom's taxonomy table of dimensions – mapping of course activities

The Knowledge Dimension	**The Cognitive Process Dimension**					
	1. Remember	***2. Understand***	***3. Apply***	***4. Analyze***	***5. Evaluate***	***6. Create***
A. Factual				#7: A.B/4.2		
B. Conceptual		#1: B.C/2.2				
C. Procedural				#2: C.B/4.3 #6: C.A/4.1	#3: C.B/5.1 #5: C.C/5.2 #8-1: C.C/5.2	
D. Metacognitive					#9: D.C/5.2 #1g: D.A/5.2	#4: D.C/6.1 #8-2: D.A/6.2 #2g: D.B/6.1

This interpretation shows that the activities are mainly located in the lower right corner of the table, which would indicate a more abstract knowledge and a higher order of thinking skills involved. While it can be a complex issue to determine the exact position of a fluid objective in not too specific content, we did try to involve abstractions, evaluations, planning, and other higher-level concepts and thinking skills. The procedural dimension, often presented in this table, might be arguable, depending on the approach to defining a "procedural" dimension. This course does not contain "strict" procedures according to the original definition, but the tasks we defined as such contain more abstract "steps" internally.

Table 2 does not represent the whole course. A visible gap in dimensions covered is in the upper left corner (factual knowledge and lower-order thinking skills), together with the empty *Apply* cognitive process column. The upper left corner with more factual knowledge is gained through self-learning with the "traditional" educational content. Such knowledge is assessed through mid-term and final exams, as it is easier to grade it more objectively. The *Apply* column is implemented in five laboratory exercises (which also include the *Create* column), where students put into practice the conceptual knowledge gained to develop the working application.

Another thing to notice is activity #1g, a synchronous group activity where students had to discuss among themselves whether and why the presented "myths/legends" were true. As this was done in the first lecture, it might seem odd that it is put in the lower right corner of the table, but this was intentional; we tried to engage students in critical abstract thinking already in the beginning. Their answers might have been partially or fully wrong, but the aim was to engage them in thinking, not to get their answers right.

In general, using the Bloom's taxonomy in planning and organizing the course content and objectives surely has a number of advantages. While it can happen that in some instances teachers might consider its application a requirement from the administration and fill out the learning objectives without implementing them, or that it is tedious to define all the needed details, it is beneficial in the long run, as both students and staff can more clearly understand the course objectives (which are sometimes quite hard to grasp depending on the course). If the envisaged outcomes and activities are presented in a meaningful way, such as using the table above, it can become clear which kinds of activities are lacking, or how is the course oriented at the moment, a bird's eye view that can't be grasped easily otherwise. If the outcomes are related to particular activities, these can also become a basis for content reuse, a concept very much needed and welcome in today's education.

5 Course evaluation

5.1 Data sources

In course evaluation, we have opted for a slightly different approach due to many surveys, polls and evaluations that our students were asked to fill out. This situation progressed in pandemic times, so we experimented with a more informal approach, having just a quick *"temperature check/how do you feel?"* question regarding some aspects of the course, which we asked every week, during the online classes. Students were responding by choosing one of the five *emojis* proposed (ranging from 1=*crying* to 5=*inlove*). In the data analysis, we translated these to 1-5 grades for easier handling but keeping in mind that this reflects students' feelings expressed in *emojis*.

Nevertheless, for some of the more critical aspects of the course, we needed some more data from the students, still trying to make it as informal as possible. Altogether, we performed our evaluation based on the following sources:

1. A short survey at the beginning of the first lecture on the reasons for course enrollment and the "temperature check" at the end of the first lecture, which included the synchronous group activity *1g: Myths and legends of open computing*.
2. An extended survey on the synchronous group activity *2g: Real-world open dataset* and the current evaluation of the course and activities, performed in week #4. The detailed results of this evaluation are available in [18].
3. A short survey with several "temperature checks" on different aspects of the course – including the question "How would you organize our course classes?", performed in the lecture week #7 (the end of the 1st lecture cycle).
4. An extended survey on activities 7, 8, 9 and the evaluation of the course and activities, performed in the lecture week #13 (end of the course).

The two short surveys and "temperature checks" were performed anonymously and synchronously using the module *JazzQuiz* in our Moodle LMS, which allowed us not only to save the feedback for detailed analysis, but also to instantly show the answers to students upon providing their feedback, and to comment on the results, in order to better explain why some activities are done in such a way, or to immediately discuss some details with the students.

The two extended surveys were also performed anonymously using the Moodle LMS, but in an asynchronous manner using the more advanced *Questionnaire* module. The questions were focused on flipped classroom activities, time spent, differences in synchronous/asynchronous and individual/group activities, in order to better understand students' preferences.

5.2 Evaluation results

At the beginning of the first lecture, we collected feedback from 18 students out of the enrolled 24 students. The main motivation of enrolled students for choosing this course was the course content (11 students, 61%), five students (27%) declared other reasons such as lack of other options for elective courses, while two students failed to state their motivation.

The initial lecture, involving a highly interactive approach to teaching, and presenting the students with a rather unexpected approach, tested their readiness and motivation to proceed in such a manner. At the end of the first lecture, the following feedback was provided using the "temperature check" polls by 17 students and presented in Table 3. Most students (15 out of 17) reacted positively to such an approach, motivating the course teachers to continue in the same manner throughout the course.

Table 3. Students' feedback after the first lecture (that was highly interactive), and at the end of the first lecture cycle

	Feedback	very negative	negative	neutral	positive	very positive
After the first lecture	***Number of students***	0	1	1	10	5
	% of responses	0%	6%	6%	59%	29%
After the first lecture cycle	***Number of students***	0	1	1	10	3
	% of responses	0%	7%	7%	67%	20%

The same type of poll was administered at the end of the first lecture cycle (lecture week #7), where 15 students provided their feedback. Comparison of the student feedback in Table 3. shows that the number of less satisfied students remained the same (two students having *neutral* and *negative* attitude), but the number of satisfied students decreased by a small number thus shifting the prevailing attitude more towards *positive* than *very positive*. However, we can conclude that the students' attitude towards the teaching approach did not change significantly in seven weeks, during the first lecture cycle.

Further analysis of "temperature check" poll answers revealed the students' attitude towards different types of teaching activities conducted during the first lecture cycle (first seven weeks of the course). As depicted in Figure 1, most of the students were very positive towards having pre-recorded lectures with weekly meetings (13 out of 15 students had a positive and very positive attitude towards pre-recorded lectures) and rejected the option of live online lectures performed in a traditional manner – just moved online (11 out of 15 students expressed a negative and very negative attitude towards live lectures). Short weekly individual assignments were very welcomed by students, strengthening their knowledge of concepts and material presented in pre-recorded lectures. Online group work, however, was not so eagerly accepted by all students, dividing their population into roughly three equal groups – ones welcomed the online group work activities (7 students), the other had no strong opinion on such a teaching method (4 students), and the third one rejected such an approach (4 students). Indeed, when asked what changes they would propose in the course implementation, 2/3 of the students concurred with the current implementation, while others expressed some minor requests for technical changes, such as earlier publishing of pre-recorded lectures and less demanding laboratory exercises.

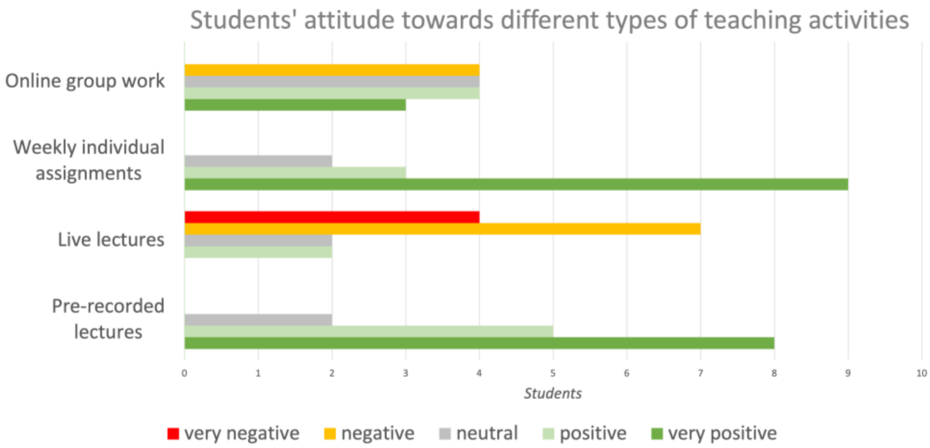

Fig. 1. Students' preferences (very negative, negative, neutral, positive, very positive) on different teaching activity types employed on the open computing course

Overall, the students' perception of the course content being interesting and useful for their future careers, collected by the poll administered at the end of the first lecture cycle (lecture week #7) and by the extended survey at the end of the course (lecture week #13), is depicted on Figure 2. Most of the students expressed a positive or strongly positive attitude towards the course being interesting and useful. Also, there are only minor differences between evaluations conducted at the end of the first lecture cycle and the end of the course. Only a small drop in the number of students that valued the course strongly positive in its usefulness can be seen between the end of the first module and the end of the course, and only one student expressed her/his overall dissatisfaction with the usefulness of the course. We can conclude that the course managed to engage students both by the approach it used and by the content it offered (more than 70% of the students found it both interesting and useful).

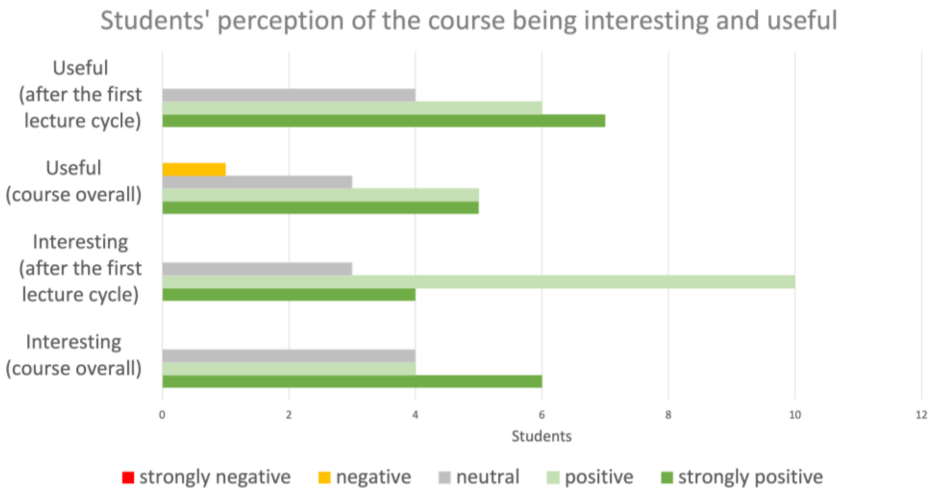

Fig. 2. Students' perception of the course content being interesting and useful

The extended survey performed at the end of the course also tested the students' attitude towards the types of activities: group synchronous activities (e.g., activity#2g: "Real-world open datasets"), individual synchronous activities (e.g., activity#9: "Speed, but caution") and a combination of individual synchronous and asynchronous activities (e.g., activity#8: "Evaluate an API!"), as described in subsection 4.1. In the survey, students suggested the desired number of activities during the course for each of the three activity types. Several conclusions can be drawn from the collected results presented in Figure 3: (*i*) students clearly prefer individual synchronous activities over the other two activity types; (*ii*) students are the least fond of individual sync/async type of activities, and this attitude is uniformly shared among all respondents; (*iii*) the suggested total number of activities during the course is around seven, implying a 1/3 decrease in the number of activities during the course, compared to the current course setup.

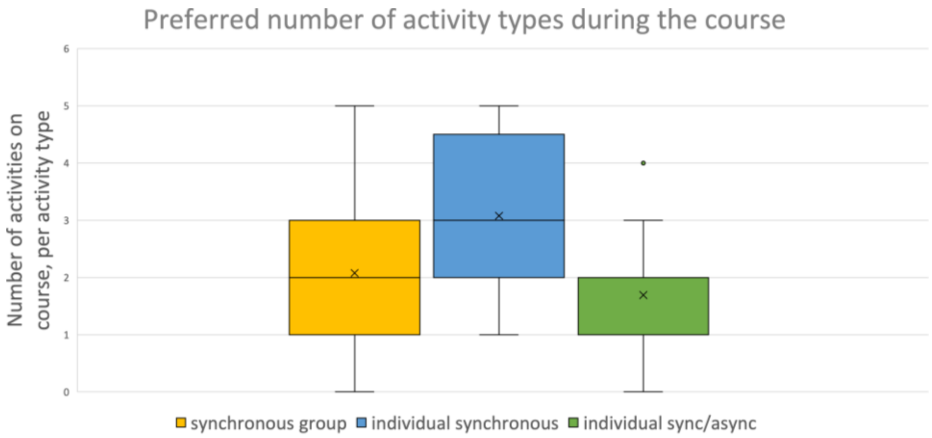

Fig. 3. Students' preferences on the number of activities and their types

6 Lessons learned

The specific lessons learned relating to particular activities are described in *Our comment* section of each activity. This section presents some of the general lessons we learned during the first implementation of the revised course, whether from our observations or students' feedback.

- Students positively react to introduced activities, evaluating the revised course as (strongly) useful and interesting;
- Students prefer individual over group activities, even in pandemic times. This might be an issue more global than related to our course, as they are not used to such work;
- Students prefer the flipped classroom approach, compared to traditional "live lectures" moved to the online setting;
- Students embrace the new course implementation, proposing only minor changes;
- Teachers see implementing active learning activities as challenging and entertaining:
- Several active learning activities require an online environment (e.g., break out rooms, posting results to a forum);
- In 2021, the time of traditional one-way teaching has gone by: students do rightly expect active learning and teaching!

In this paper, our learning objectives and activities were fit to revised Bloom's taxonomy in 1:1 mapping, defining only one – most prominent – knowledge and cognitive process dimension per objective and activity. However, an activity often comprises more Bloom's taxonomy elements (as seen in two-staged activity #8). Further fine-tuning of activities, objectives and taxonomy dimensions should be performed, after another year of carrying out the course and gaining more experiences and activity feedback.

7 Conclusion and future work

This paper presents a revised version of the *Open Computing* course and its distinguishing feature of real-world mini-activities to be used in a flipped classroom model.

The most prominent changes for newly developed course were including small weekly activities on "real-world" situations, performed either individually or in groups, in a synchronous or asynchronous manner. Asynchronous tasks were accomplished in our LMS, while all tasks were at least discussed - or fully carried out – during the online joint weekly meetings. The student evaluation of the course shows that students find this course useful and interesting, preferring individual assignments over group ones. Still, they concur with our vision of this course, proposing some minor changes.

While the teachers were very satisfied with the new approach, a set of fine-tuning and minor revisions needs to be applied in the further academic years to better fit the activities to our course objectives, based on students' feedback and our observations.

An essential part of the future work would be finding a good way to reuse the flipped classroom method and described activities in our course once the COVID-19 pandemic ends. Moving back to the classroom will bring additional organization issues in activities which are very easily done online, such as quick forming of breakout rooms for discussion, easy screen sharing or collaborating. It is possible that some activities should be revised for in-classroom use, or that a hybrid method and partial in-class teaching would become available to us. We believe that one good consequence of the pandemic will be at least the partial move from traditional lesson setup to modern approaches.

Although focused primarily on open concepts in their technical context, this list of activities could also be attractive to a broader audience, showing how real-world context could be applied in other, non-related courses. While the topics of interest and activities would be different, ideas and methods of real-world flipped classroom concept would continue to be useful in other teaching contexts. Therefore, we hope our experiences could be helpful for other higher education teaching staff.

8 Acknowledgment

We would like to thank all students of *Open Computing* course, especially those who enrolled in the year 2020/21, for helping us to create a better course. We would like to thank prof. Mario Žagar, who introduced the concept of open computing to our Faculty and started this course many years ago. This research is part of the Twinning Open Data Operational project that has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement no. 857592.

9 References

[1] J. Willinsky, "The unacknowledged convergence of open source, open access, and open science," *First Monday*, vol. 10, no. 8. 2005. https://doi.org/10.5210/fm.v10i8.1265

[2] P. Mirowski, "The future(s) of open science," *Soc. Stud. Sci.*, vol. 48, no. 2, 2018. https://doi.org/10.1177/0306312718772086

[3] *Open innovation, open science, open to the world: a vision for Europe*. European Commission and DG for Research and Innovation, 2016.

[4] T. Coughlan, "The use of open data as a material for learning," *Educ. Technol. Res. Dev.*, vol. 68, no. 1, 2020. https://doi.org/10.1007/s11423-019-09706-y

[5] C. K. Pereira, S. W. M. Siqueira, B. P. Nunes, and S. Dietze, "Linked Data in Education: A Survey and a Synthesis of Actual Research and Future Challenges," *IEEE Trans. Learn. Technol.*, vol. 11, no. 3, pp. 400–412, Jul. 2018. https://doi.org/10.1109/TLT.2017.2787659

[6] A. Mansourian *et al.*, "Showcase of Existing Active Teaching Practices." SPIDER, 2021.

[7] F. Ozdamli and G. Asiksoy, "Flipped Classroom Approach," *World J. Educ. Technol. Curr. Issues*, vol. 8, no. 2, pp. 98–105, Jul. 2016. https://doi.org/10.18844/wjet.v8i2.640

[8] A. King, "From Sage on the Stage to Guide on the Side," *Coll. Teach.*, vol. 41, no. 1, 1993. https://doi.org/10.1080/87567555.1993.9926781

[9] L. Abeysekera and P. Dawson, "Motivation and cognitive load in the flipped classroom: definition, rationale and a call for research," *High. Educ. Res. Dev.*, vol. 34, no. 1, 2015. https://doi.org/10.1080/07294360.2014.934336

[10] I. T. Awidi and M. Paynter, "The impact of a flipped classroom approach on student learning experience," *Comput. Educ.*, vol. 128, 2019. https://doi.org/10.1016/j.compedu.2018.09.013

[11] E. Mazur and M. D. Somers, "Peer Instruction: A User's Manual," *Am. J. Phys.*, vol. 67, no. 4, 1999. https://doi.org/10.1119/1.19265

[12] S. B. Velegol, S. E. Zappe, and E. Mahoney, "The evolution of a flipped classroom: Evidence-based recommendations," *Adv. Eng. Educ.*, vol. 4, no. 3, 2015. https://doi.org/10.1109/ECONF.2015.34

[13] H. Hendrik and A. Hamzah, "Flipped Classroom In Programming Course: A Systematic Literature Review," *Int. J. Emerg. Technol. Learn.*, vol. 16, no. 2, 2020. https://doi.org/10.3991/ijet.v16i02.15229

[14] E. M. Bakheet and A. M. Gravell, "Would Flipped Classroom be My Approach in Teaching Computing Courses: Literature Review," 2021. https://doi.org/10.1109/ICIET51873.2021.9419631

[15] M. Lou Maher, C. Latulipe, H. Lipford, and A. Rorrer, "Flipped classroom strategies for CS education," 2015. https://doi.org/10.1145/2676723.2677252

[16] L. Anderson and D. Krathwohl, "A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives," 2001, Accessed: Jul. 21, 2021. [Online]. Available: https://eduq.info/xmlui/handle/11515/18345

[17] L. Richardson, "Justice Will Take Us Millions Of Intricate Moves," *QCon conference*. Nov. 20, 2008.

[18] I. Bosnic, I. Cavrak, and A. Zuiderwijk, "Introducing Open Data Concepts to STEM Students Using Real-World Open Datasets," in *44rd MIPRO 2021 International Convention - Proceedings*, 2021, pp. 1761–1766. https://doi.org/10.23919/MIPRO52101.2021.9596998

10 Authors

Ivana Bosnić works as assistant professor, at the University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Control and Computer Engineering, Unska 3, Zagreb, Croatia (email: ivana.bosnic@fer.hr).

Igor Čavrak works as associate professor, at the University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Control and Computer Engineering, Unska 3, Zagreb, Croatia (email: igor.cavrak@fer.hr).

Bastiaan van Loenen works as associate professor at Delft University of Technology, Department of Architecture and the Built Environment, Delft, the Netherlands (email: b.vanloenen@tudelft.nl).

Article submitted 2021-07-26. Resubmitted 2022-01-24. Final acceptance 2022-01-24. Final version published as submitted by the authors.