DISTINGUISHED EDUCATION EXCELLENCE AWARDS

Category: Professional Practice Innovation Award

Case Study

Title: Active Learning in Systems Software Education (ALISSE) Framework

Netzahualcoyotl Hernandez-Cruz, Samuel Moore and Paul McCullagh

Summary:

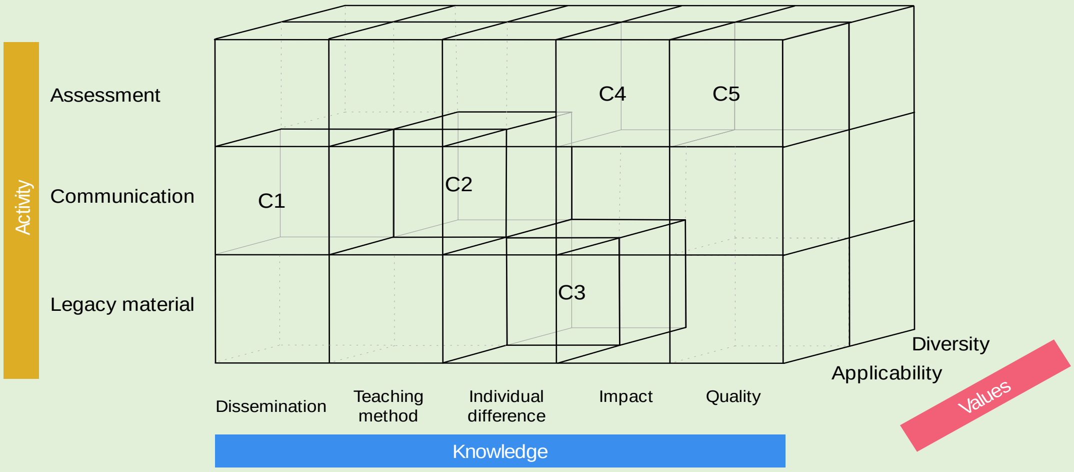
Active Learning in Systems Software Education (ALISSE[1]) is a framework tailored for designing computing teaching material. ALISSE was used to redesign laboratory material in a Year 1 Systems Software module. The case study showcases the delivery with synchronous tutor support to a large cohort (N=225), coupled with a staged online assessment encouraging exploratory investigation to promote active learning. Monitored average attendance shows an increase in the students' engagement from 47% (2019-20) to 80% (2020-21), enhancing the quality of the student learning experience in unprecedented challenging times. Feedback was delivered upon test completion to allow students to reapply effort to succeed during module delivery. The assessment facilitated the adoption of in-module assessment and recovery (IMAR) to boost continued engagement with eight students benefiting. ALISSE has positively impacted the students' access and helped in their academic formation towards industrial placement preparation in Year 2. In addition and to showcase the transferability of ALISSE, the module was subsequently delivered to the Ulster College in Xi'an, China (SUST), where the staff members expressed its usefulness and effectiveness in supporting students learning.

What was done:

Principles and guidelines for developing social science and linguistics teaching material have been reported [1, 2]. To extend this pedagogic theory to computing, we developed ALISSE. This framework implements five relevant constructors to facilitate engaging teaching material with communication based on Shneiderman's interface design rules [3], see Figure 1. ALISSE's constructors were empirically selected based on the authors' teaching experience; however, additional constructors could be included in future versions. The framework is inspired by the three dimensions of the UK Professional Standards Framework [4] and embraces the active learning technique defined as "the activities that students do to construct knowledge and understanding" [5].

ALISSE underpins the Five and Fifty strategic plan [6] by addressing a core area of Ulster's teaching excellence strategy: developing an innovative curriculum, using contemporary methods of pedagogy that foster diversity, differentiation, and increased opportunities for access.

[1] Alisse is a name of Greek origin meaning "rational".



C1 –Communication, Dissemination, and Applicability

This constructor develops the appealing and stylish visuals for representing academic material, along with the dissemination method, so students experience the applicability of the acquired skills to solve practical problems.

Guideline units:

- Would the student-experience benefit from integrating multimedia resources to the teaching material?
- Do the exercises test the student's skills to solve real-life problems?
- Does the content of the material organise hierarchically?

C2 –Communication, Teaching method, and Diversity

This constructor caters to differences among student's learning abilities, and focuses on providing support to strength their confidence and independence.

Guideline units:

- Does the material implements UX accessibility recommendations?
- Are the navigation and interaction across the different sections intuitive to the student?

C3 –Legacy material, Individual difference, and Diversity

This constructor points towards learning from previous teaching material. It suggests to investigate what practices have been applied in the past, adopt what has been of use and improve what hasn't. Unlike C2, this constructor focuses on the difference in the students expertise to level up the material expectation.

Guideline units:

- Does the material consider the different levels of expertise of the students?

C4 –Assessment, Impact, and Applicability

This constructor addresses the workload of a lecture. It suggests a schema, in which a lesson is split into sequentially placed building blocks to ensure the expected knowledge is learned.

Guideline units:

- Does the material split into small sections to distribute the teaching material?

C5 –Assessment, Quality and Applicability

This constructor highlights the importance of monitoring the teaching material considering the social, and pedagogical changes across student generations as well as the development of new teaching technology.

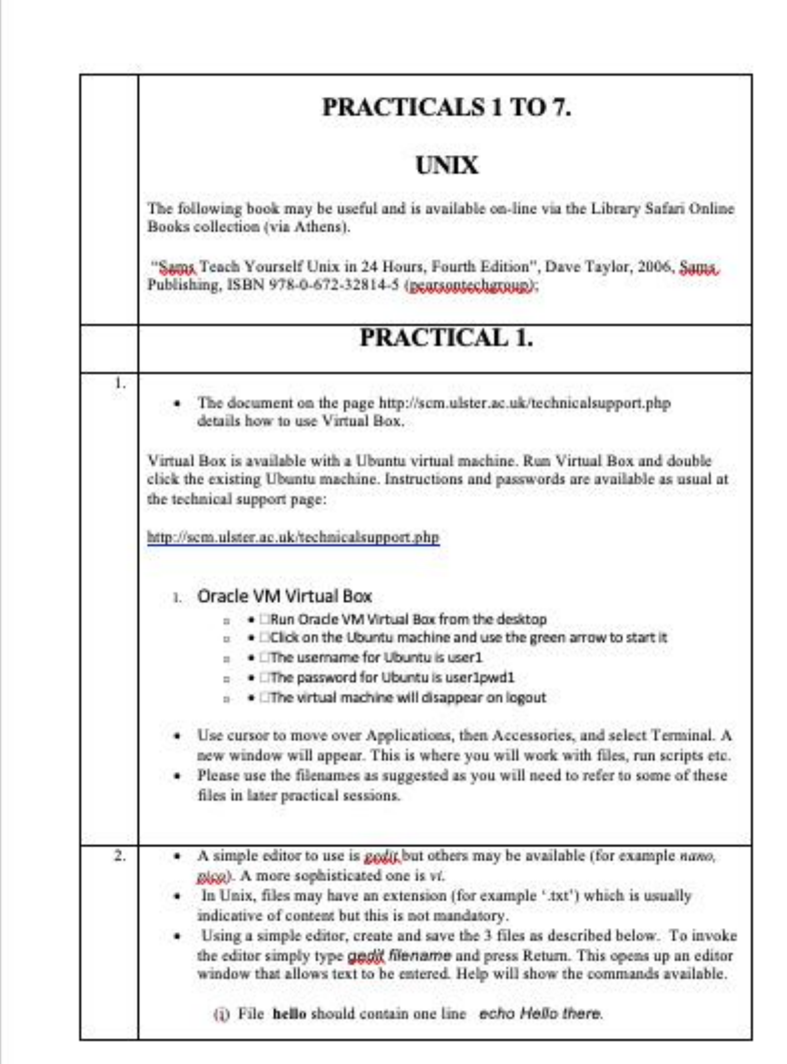
Guideline units:

- Does the teaching staff has considered an approach to monitor the student's engagement to the learning experience?

Figure 1 – ALISSE framework illustrating the contribution of pedagogic activities, knowledge consideration and value aspects as the foundation to each of the five constructors of the framework. Guideline units illustrate a suggestion for implementing the constructor into the lab material.

Motivation and aims:

Prior to the 2021 delivery, the lab material required self-paced completion of a logbook (Figure 2), which could be completed without significant scheduled engagement. Average attendance in 2019-2020 was 47%[2], highlighting the missing value of engagement in practice and research. Hence, there was a need for promoting better student engagement.



PRACTICALS 1 TO 7.

UNIX

The following book may be useful and is available on-line via the Library Safari Online Books collection (via Athens).

"Sams Teach Yourself Unix in 24 Hours, Fourth Edition", Dave Taylor, 2006, Sams Publishing, ISBN 978-0-672-32814-5 (pearsontechgroup);

PRACTICAL 1.

1.

- The document on the page http://scm.ulster.ac.uk/technicalsupport.php details how to use Virtual Box.

Virtual Box is available with a Ubuntu virtual machine. Run Virtual Box and double click the existing Ubuntu machine. Instructions and passwords are available as usual at the technical support page:

http://scm.ulster.ac.uk/technicalsupport.php

1. Oracle VM Virtual Box
   - □Run Oracle VM Virtual Box from the desktop
   - □Click on the Ubuntu machine and use the green arrow to start it
   - □The username for Ubuntu is user1
   - □The password for Ubuntu is user1pwd1
   - □The virtual machine will disappear on logout

- Use cursor to move over Applications, then Accessories, and select Terminal. A new window will appear. This is where you will work with files, run scripts etc.
- Please use the filenames as suggested as you will need to refer to some of these files in later practical sessions.

2.

- A simple editor to use is *gedit* but others may be available (for example *nano*, *pico*). A more sophisticated one is *vi*.
- In Unix, files may have an extension (for example '.txt') which is usually indicative of content but this is not mandatory.
- Using a simple editor, create and save the 3 files as described below. To invoke the editor simply type *gedit filename* and press Return. This opens up an editor window that allows text to be entered. Help will show the commands available.

(i) File **hello** should contain one line *echo Hello there.*

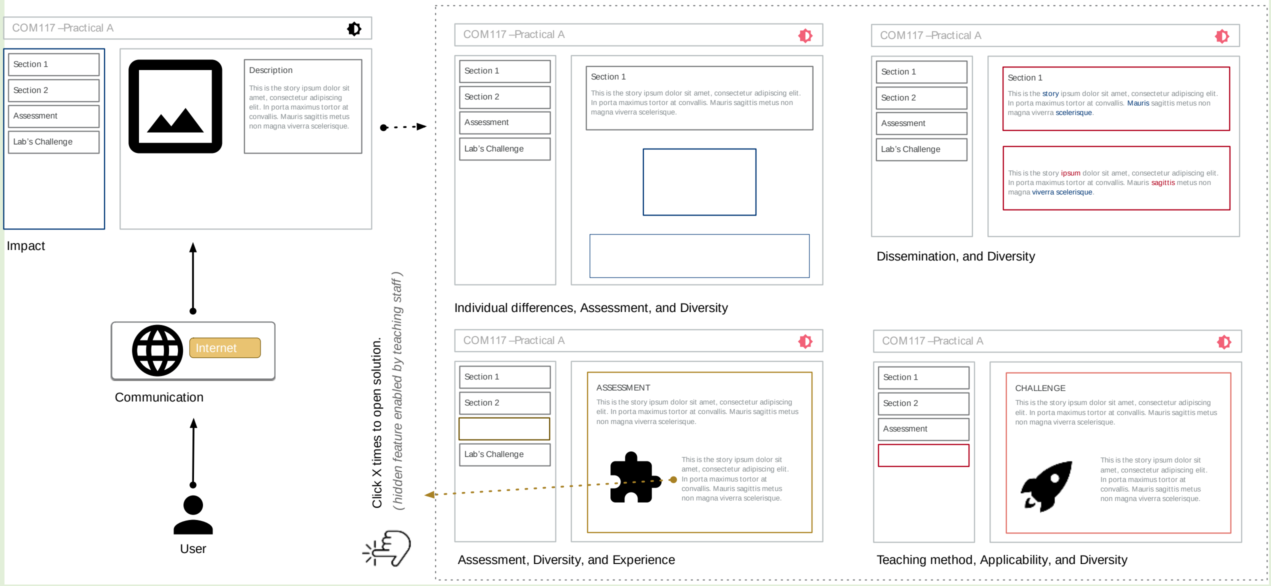
Figure 2 –Legacy logbook.

The ALISSE redesign adopted a new approach that provided interactive learning coupled with shorter staged assessment components and an environment with which the students would be keen to engage. The aim was to promote deeper learning of core computational skills, which could be assessed interactively with expedited feedback, to ensure that students remained on course. IMAR was adopted as our approach facilitated catch-up for any student who may have fallen behind (e.g. due to illness).

Implementation:

[2] This figure, as reported by Blackboard, may be subject to error due to face to face monitoring but is representative of our experience and was a key motivation.

The implementation of the ALISSE constructors is represented in Figure 3. The pedagogical material was supported by innovative research and computing practice. The lab experience profited from PhD students and Computing officers who acted as tutors and trouble-shooters. This provided helpful preparation for students seeking industrial placement during Year 2.


COM117 –Practical A
Section 1
Section 2
Assessment
Lab's Challenge
Description
Impact
Internet
Communication
User
Click X times to open solution.
(hidden feature enabled by teaching staff)
Individual differences, Assessment, and Diversity
Dissemination, and Diversity
ASSESSMENT
CHALLENGE
Assessment, Diversity, and Experience
Teaching method, Applicability, and Diversity

C1 –Communication, Dissemination, and Applicability

The material was designed to be entirely online, using a minimalistic but refreshing graphical design. The formality of the Lab-book was updated to stylish multimedia interaction and hyperlinks to external resources. This technology was chosen as it put together interactive resources for the students to engage.

C2 –Communication, Teaching method, and Diversity

To stimulate engagement, each section contained problem-solving exercises. These increased student confidence, allowing them to reach independence by experiencing the benefit of applying the learned knowledge.

A friendly vocabulary was adopted to connect with students in a collegiate way. Design elements were enabled such as: light and night mode to adjust to an individual's reading comfort.

C3 –Legacy material, Individual difference, and Diversity

The existing lab material consisted of an uninspiring Log-book. Nevertheless, the core information, number of topics and their scope was relevant as a reference for planning the new material.

We addressed diversity by providing challenge to the advanced student while facilitating assistance to others. Multimedia provided step by step guidance. Such resources were available, yet, activated only when the student requested them; this way, the advanced student wasn't disturbed by the unneeded information.

C4 –Assessment, Impact, and Applicability

The material consisted of a series of carefully designed building blocks, where each practical paved the foundation to the next, to ensured that relevant information was attempted (understood), as the student progressed.

For each topic, the material was designed as sections/sub-sections. The relevant sub-section included formative assessment to ensure the learning of specific pieces of information before moving forwards. Summative assessment comprised three short on-line computer marked tests.

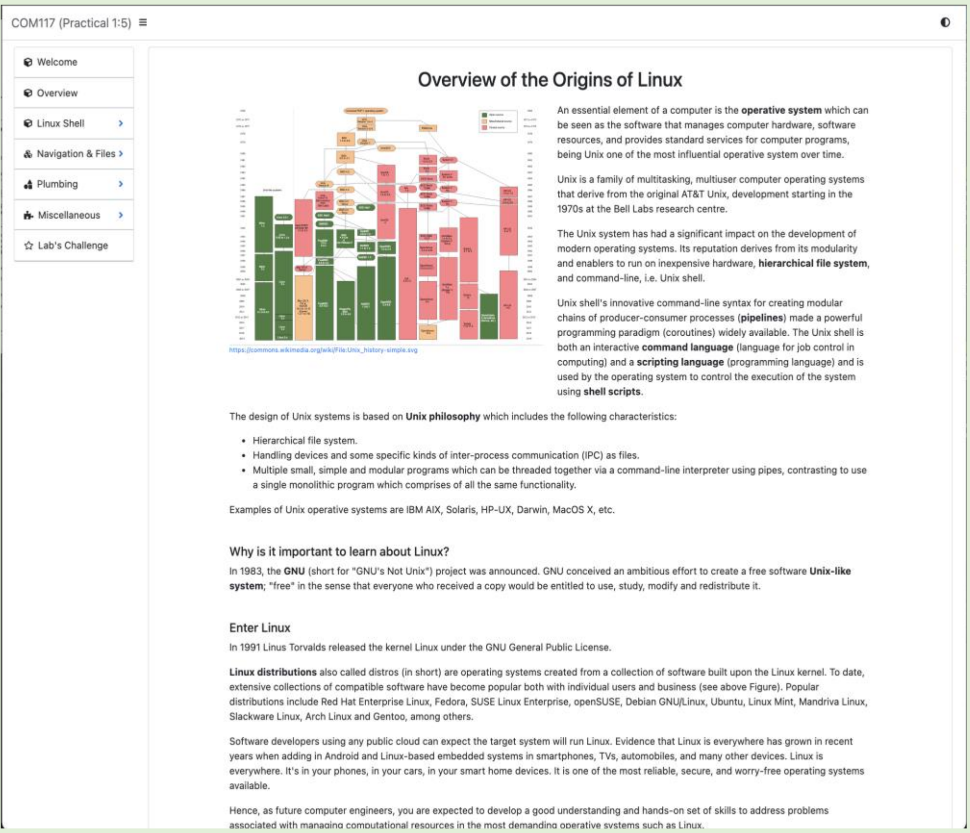
Applicability was addressed by establishing problems in which students were asked to analysing data from the Covid pandemic.

C5 –Assessment, Quality and Applicability

The lab material was developed and monitored by three teaching staff and validated by two external teaching staff before deployment. Furthermore, the modular design facilitated adjustments while deployed.

Figure 3 – Implementation of ALISSE.

Figures 4 - 6 shows exemplars of the updated lab environment.



Overview of the Origins of Linux

https://commons.wikimedia.org/wiki/File:Unix_history-simple.svg

An essential element of a computer is the **operative system** which can be seen as the software that manages computer hardware, software resources, and provides standard services for computer programs, being Unix one of the most influential operative system over time.

Unix is a family of multitasking, multiuser computer operating systems that derive from the original AT&T Unix, development starting in the 1970s at the Bell Labs research centre.

The Unix system has had a significant impact on the development of modern operating systems. Its reputation derives from its modularity and enablers to run on inexpensive hardware, **hierarchical file system**, and command-line, i.e. Unix shell.

Unix shell's innovative command-line syntax for creating modular chains of producer-consumer processes (**pipelines**) made a powerful programming paradigm (coroutines) widely available. The Unix shell is both an interactive **command language** (language for job control in computing) and a **scripting language** (programming language) and is used by the operating system to control the execution of the system using **shell scripts**.

The design of Unix systems is based on **Unix philosophy** which includes the following characteristics:

- Hierarchical file system.
- Handling devices and some specific kinds of inter-process communication (IPC) as files.
- Multiple small, simple and modular programs which can be threaded together via a command-line interpreter using pipes, contrasting to use a single monolithic program which comprises of all the same functionality.

Examples of Unix operative systems are IBM AIX, Solaris, HP-UX, Darwin, MacOS X, etc.

Why is it important to learn about Linux?

In 1983, the **GNU** (short for "GNU's Not Unix") project was announced. GNU conceived an ambitious effort to create a free software **Unix-like system**; "free" in the sense that everyone who received a copy would be entitled to use, study, modify and redistribute it.

Enter Linux

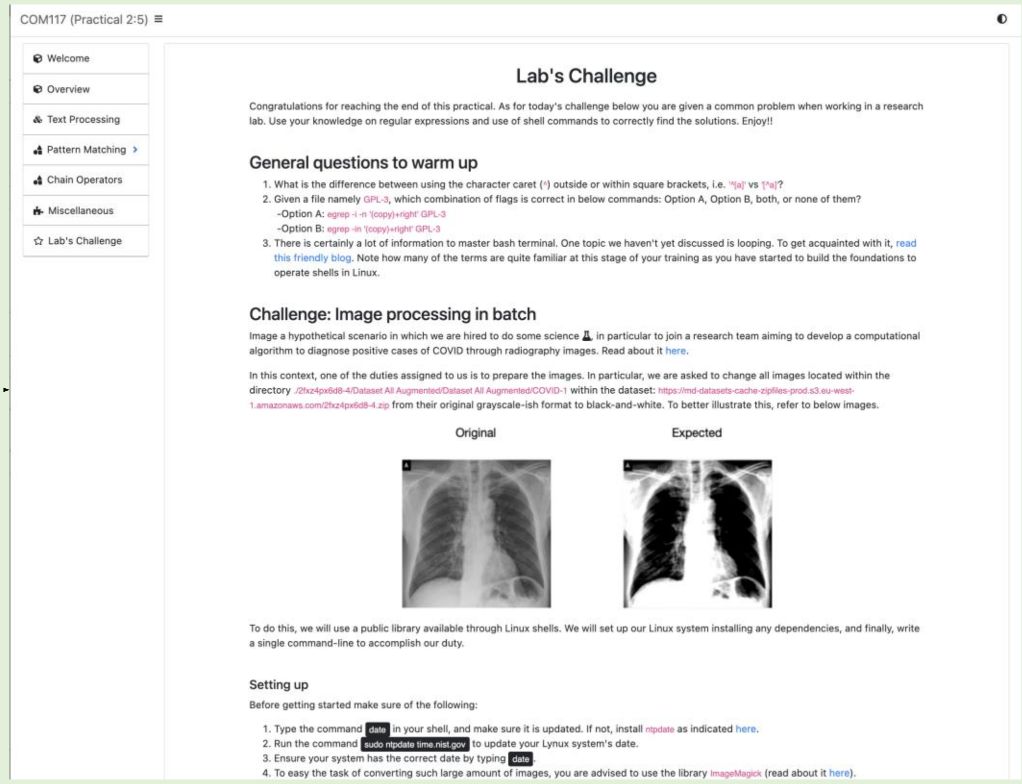
In 1991 Linus Torvalds released the kernel Linux under the GNU General Public License.

Linux distributions also called distros (in short) are operating systems created from a collection of software built upon the Linux kernel. To date, extensive collections of compatible software have become popular both with individual users and business (see above Figure). Popular distributions include Red Hat Enterprise Linux, Fedora, SUSE Linux Enterprise, openSUSE, Debian GNU/Linux, Ubuntu, Linux Mint, Mandriva Linux, Slackware Linux, Arch Linux and Gentoo, among others.

Software developers using any public cloud can expect the target system will run Linux. Evidence that Linux is everywhere has grown in recent years when adding in Android and Linux-based embedded systems in smartphones, TVs, automobiles, and many other devices. Linux is everywhere. It's in your phones, in your cars, in your smart home devices. It is one of the most reliable, secure, and worry-free operating systems available.

Hence, as future computer engineers, you are expected to develop a good understanding and hands-on set of skills to address problems associated with managing computational resources in the most demanding operative systems such as Linux.

Figure 4 – Interface promoting accessibility design.



Lab's Challenge

Congratulations for reaching the end of this practical. As for today's challenge below you are given a common problem when working in a research lab. Use your knowledge on regular expressions and use of shell commands to correctly find the solutions. Enjoy!!

General questions to warm up

1. What is the difference between using the character caret (^) outside or within square brackets, i.e. '^[a]' vs '[^a]'?
2. Given a file namely GPL-3, which combination of flags is correct in below commands: Option A, Option B, both, or none of them?
   -Option A: egrep -i -n '(copy)+right' GPL-3
   -Option B: egrep -in '(copy)+right' GPL-3
3. There is certainly a lot of information to master bash terminal. One topic we haven't yet discussed is looping. To get acquainted with it, read this friendly blog. Note how many of the terms are quite familiar at this stage of your training as you have started to build the foundations to operate shells in Linux.

Challenge: Image processing in batch

Image a hypothetical scenario in which we are hired to do some science ⚗, in particular to join a research team aiming to develop a computational algorithm to diagnose positive cases of COVID through radiography images. Read about it here.

In this context, one of the duties assigned to us is to prepare the images. In particular, we are asked to change all images located within the directory ./2fxz4px6d8-4/Dataset All Augmented/Dataset All Augmented/COVID-1 within the dataset: https://md-datasets-cache-zipfiles-prod.s3.eu-west-1.amazonaws.com/2fxz4px6d8-4.zip from their original grayscale-ish format to black-and-white. To better illustrate this, refer to below images.

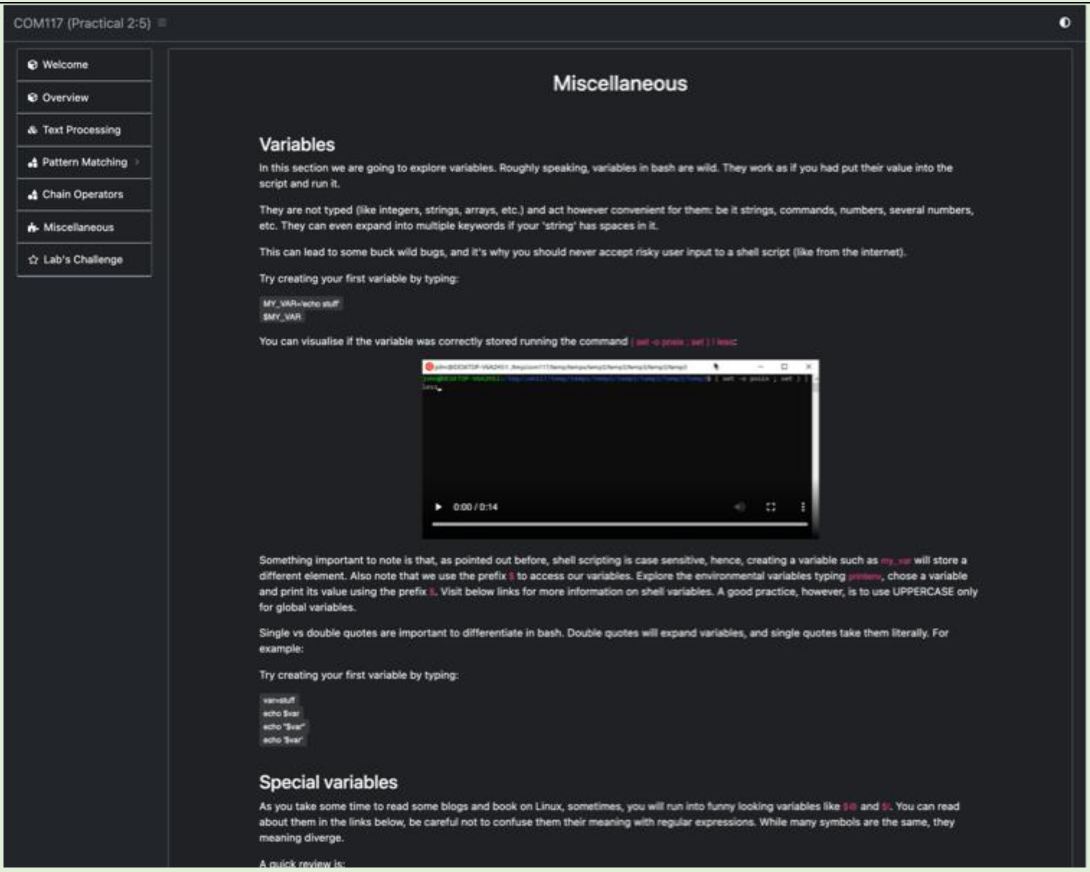
To do this, we will use a public library available through Linux shells. We will set up our Linux system installing any dependencies, and finally, write a single command-line to accomplish our duty.

Setting up

Before getting started make sure of the following:

1. Type the command `date` in your shell, and make sure it is updated. If not, install ntpdate as indicated here.
2. Run the command `sudo ntpdate time.nist.gov` to update your Lynux system's date.
3. Ensure your system has the correct date by typing `date`.
4. To easy the task of converting such large amount of images, you are advised to use the library ImageMagick (read about it here).

Figure 5 – Problem-solving approach to test the students' skills and facilitate formative feedback.



Miscellaneous

Variables

In this section we are going to explore variables. Roughly speaking, variables in bash are wild. They work as if you had put their value into the script and run it.

They are not typed (like integers, strings, arrays, etc.) and act however convenient for them: be it strings, commands, numbers, several numbers, etc. They can even expand into multiple keywords if your 'string' has spaces in it.

This can lead to some buck wild bugs, and it's why you should never accept risky user input to a shell script (like from the internet).

Try creating your first variable by typing:

```
MY_VAR='echo stuff'
$MY_VAR
```

You can visualise if the variable was correctly stored running the command (set -o posix ; set) | less:

Something important to note is that, as pointed out before, shell scripting is case sensitive, hence, creating a variable such as my_var will store a different element. Also note that we use the prefix $ to access our variables. Explore the environmental variables typing printenv, chose a variable and print its value using the prefix $. Visit below links for more information on shell variables. A good practice, however, is to use UPPERCASE only for global variables.

Single vs double quotes are important to differentiate in bash. Double quotes will expand variables, and single quotes take them literally. For example:

Try creating your first variable by typing:

```
var=stuff
echo $var
echo "$var"
echo '$var'
```

Special variables

As you take some time to read some blogs and book on Linux, sometimes, you will run into funny looking variables like $@ and $!. You can read about them in the links below, be careful not to confuse them their meaning with regular expressions. While many symbols are the same, they meaning diverge.

A quick review is:

Figure 6 – Use of colour contrast and multimedia features to provide 'how to' added value.

Figure 7 provides a comparison from 2019-20 to 2020-21. The online approach replaced the asynchronous self-paced logbook with a staged 'just in time' assessment (3 components with 2 IMAR components). The student cohort was split into two groups, supported by the lab leader, four tutors and module lecturers. The assessment was delivered as a Blackboard quiz from a question bank, ensuring that students received varied but equivalent assessments.

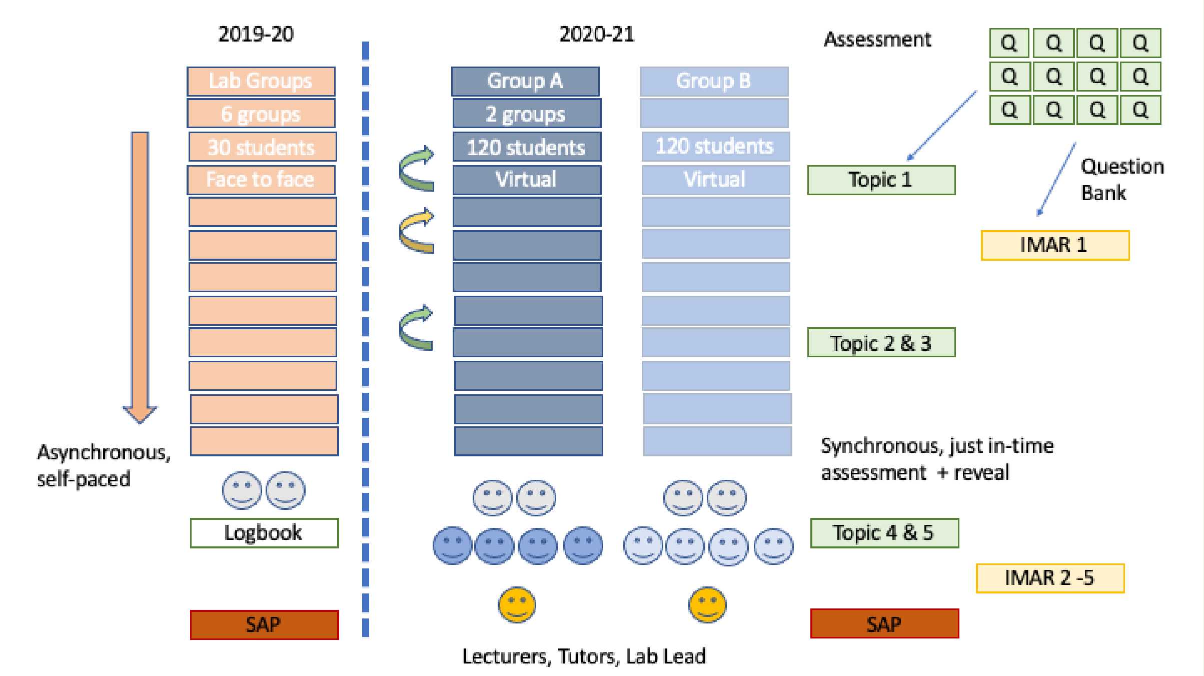


Figure 7 – Comparison of delivery and assessment strategies for 2019-20 and 2020-21.

In the 2019-20 module, the performance lacked differentiation, and engagement monitored by attendance was of concern at 47%. Figure 8 illustrates 2020-21 module performance versus attendance for students who completed the module. There is a positive correlation (*R=0.51),* and the average attendance for the module was 80% (sd=23%). The average module mark was 62% (sd=18%), with good differentiation.

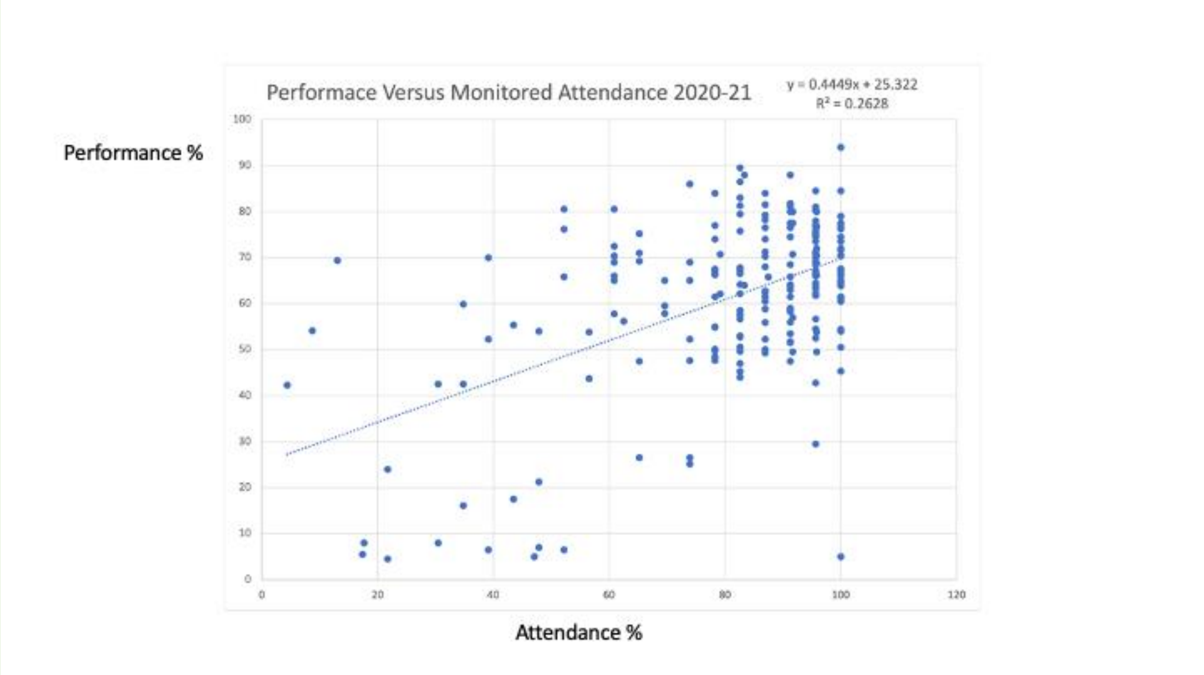


Figure 8 –Performance versus Attendance 2020-21 (N=213, Slope =0.44).

Figure 9 illustrates the breakdown of module retakes against module for Year 1. Systems Software is M5. Completion compares favourably with the other modules, with the same number of retakes as M3. Interestingly M3 also offered a similar assessment strategy based on short online class tests.

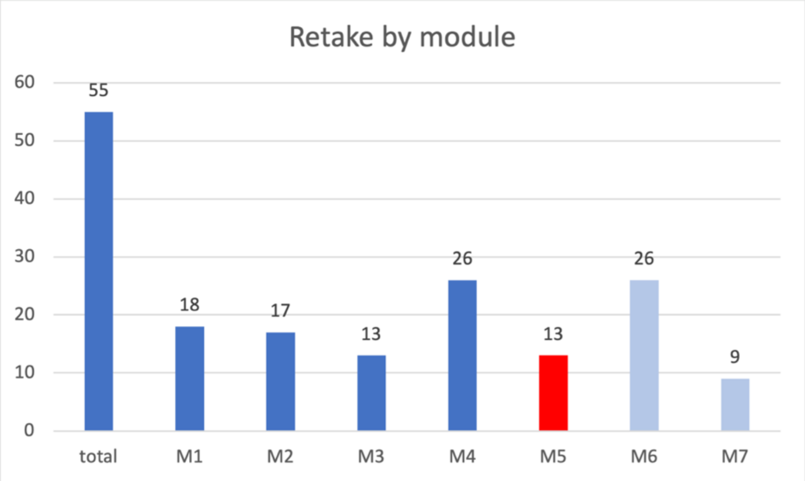


Figure 9 –Retake opportunity of Year 1 cohort by modules.

Of the 13 students who were offered a retake opportunity, all students succeeded, which shows the success of our approach. 8 students had benefited from IMAR, and only 1 student completed during the Supplementary Assessment Period.

In Student Staff Consultation Committee, no negative issues were raised with module delivery. Formal student feedback is based on only 22/225 responses [7]. The feedback is predominantly positive; see Figures 10 and 11, where most answers to 11/12 questions were 'Strongly Agree' plus 'Agree', i.e. positive sentiment.

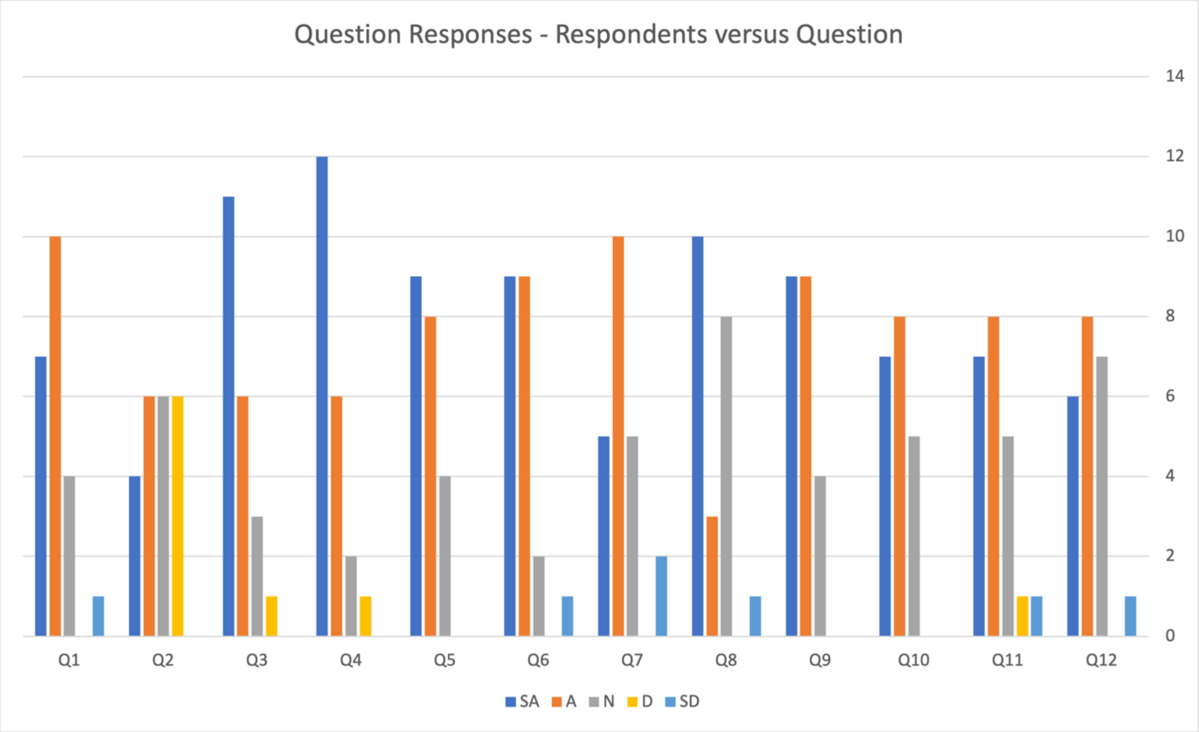
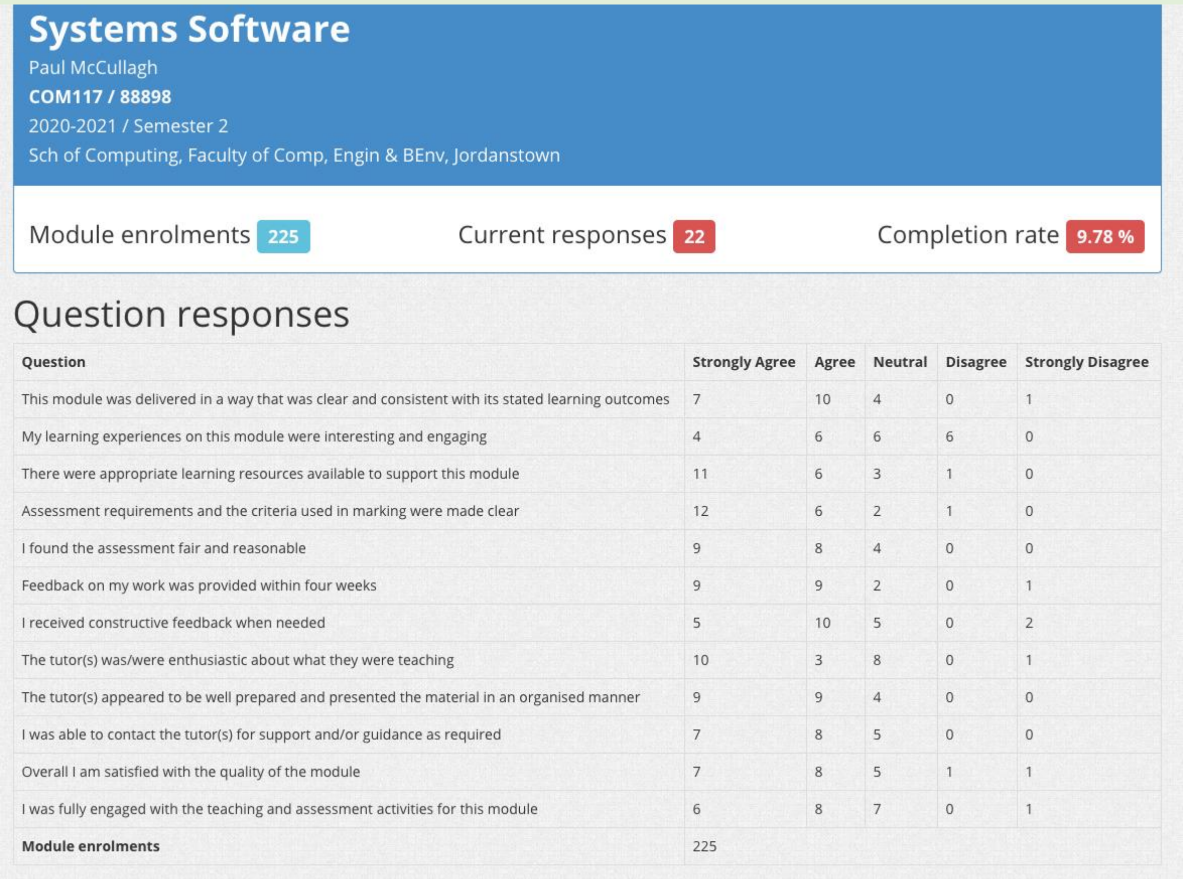


Figure 10 – Student Questionnaire metrics.



Systems Software

Paul McCullagh

COM117 / 88898

2020-2021 / Semester 2

Sch of Computing, Faculty of Comp, Engin & BEnv, Jordanstown

Module enrolments 225

Current responses 22

Completion rate 9.78 %

Question responses

Question	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
This module was delivered in a way that was clear and consistent with its stated learning outcomes	7	10	4	0	1
My learning experiences on this module were interesting and engaging	4	6	6	6	0
There were appropriate learning resources available to support this module	11	6	3	1	0
Assessment requirements and the criteria used in marking were made clear	12	6	2	1	0
I found the assessment fair and reasonable	9	8	4	0	0
Feedback on my work was provided within four weeks	9	9	2	0	1
I received constructive feedback when needed	5	10	5	0	2
The tutor(s) was/were enthusiastic about what they were teaching	10	3	8	0	1
The tutor(s) appeared to be well prepared and presented the material in an organised manner	9	9	4	0	0
I was able to contact the tutor(s) for support and/or guidance as required	7	8	5	0	0
Overall I am satisfied with the quality of the module	7	8	5	1	1
I was fully engaged with the teaching and assessment activities for this module	6	8	7	0	1
Module enrolments	225				

Figure 11 – Student Feedback.

Transferability:

The module was subsequently delivered to students at the Ulster College in Xi'an, China. This partnership brings together the teaching expertise of UU and SUST [8]. Students of the SUST cohort had not been familiarised with Python at the point of module delivery; therefore, the module was tailored to deliver elements in C programming, which was a more appropriate tool for that cohort.

Further information:

Head of School at UU, Chris Nugent, commented
“The format of the labs was extremely helpful in delivering the module. Given the remote delivery to the cohort in China, we were able to use the lab sheets to both structure the operation of the labs and provide students with all the necessary information they required in a step-by-step manner. The inclusion of theory, practical class work and self-assessment also worked well and again was extremely well suited to the remote delivery of the module content.”

Lecturer at UU, Shuai Zhang, commented
“The performance from the students was great, especially the assessment related to the lab/practical skills. I think it has demonstrated that the usefulness and effectiveness of the lab materials in supporting students learning.”
“Miss Hua Wu –our SUST colleague has been teaching the subject areas for quite several years. She has also commented on the flexibility of the lab materials and that the material is easy for the students to follow and engage. She had asked how the materials had been prepared for possible adoption in her teaching.”

Teaching supportive staff at UU commented
During module interaction, students provided positive comments, “instructions and exercises were clear”, “module tested my technical skills”, “Linux code helped me to envision applicability to solve real-life problems”, and several students benefited directly from the flexible rollout of IMAR.

Lecturer at SUST, Hua Wu, commended
“This module went well in SUST this semester. The results of the experiment are very good, and the students are quite satisfied. The students participated in the experimental module throughout. The experiment instructions were very detailed. The students were satisfied with the content of the experiment module and met the requirements of the experiment.”

Resources:
A complementary description and courtesy sample of ALISSE framework implementation is available at:
https://ulster-my.sharepoint.com/:f:/g/personal/pj_mccullagh_ulster_ac_uk/ElckIm76C6pOjXAQWcBd7X8B-2hTXVgiz3oxQGMD2cltGg?e=ghNBmZ

References

1. Jolly, D. & Bolitho, R. (2011) A framework for materials writing. In: Tomlinson, B. (ed). Materials Development in Language Teaching. (2nd ed) Cambridge: Cambridge University Press. pp.107- 134.
2. Tomlinson, B. (2012) Materials development for language learning and teaching. Language Teaching 45 (02): pp. 143-179.
3. Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., and Elmqvist, N., Designing the User Interface: Strategies for Effective Human-Computer Interaction: Sixth Edition, Pearson (May 2016) http://www.cs.umd.edu/hcil/DTUI6, accessed July 2021
4. Framework, https://www.advance-he.ac.uk/knowledge-hub/uk-professional-standards-framework-ukpsf, accessed July 2021.
5. Brame, C.J. Active Learning What is it? What's the theoretical basis? https://cft.vanderbilt.edu/wp-content/uploads/sites/59/Active-Learning.pdf, accessed July 2021.
6. Five and Fifty, https://www.ulster.ac.uk/fiveandfifty/strategicplan.pdf, accessed July 2021.
7. Module survey http://feedback.ulster.ac.uk/dashboard/view/id/88898, accessed July 2021.
8. Ulster University partners with Shaanxi University of Science and Technology to establish Ulster College in Northern China https://www.ulster.ac.uk/news/2019/september/ulster-university-partners-with-shaanxi-university-of-science-and-technology-to-establish-ulster-college-in-northern-china, accessed July 2021.