Learning and Teaching Enhancement Project Report, June, 2015

Project Title: "Helping 1st Year CS Students to Become Independent Learners Through Automated Feedback"

School & department: School of Mathematical and Computer Sciences

Please be aware that this report will be publicly available on the University's webpages
Please complete in Word and return to Mirren.McLeod@hw.ac.uk (reports not completed in Word will be returned)

1	***Names and Heriot-Watt University contact details of project team (please identify the project lead/ report author):*** Gudmund Grov (lead author), Graeme Reid (project student) and Verena Rieser (project lead/ on maternity leave)
2	***Key words:*** *Automatic feedback, 1st year students, large classes, computer programming, independent learners*
3	***The problem being addressed, with background and context:*** This project is concerned with the problem of how to provide timely and constructive feedback to first year Computer Science (CS) students by developing an automated feedback framework. Learning how to program computers is one of the major hurdles for our first year students across Heriot-Watt campuses. A multi-national, multi-institutional assessment of programming skills of first-year CS students found that many students do not know how to program at the conclusion of their introductory courses (McCracken et al., 2001). Some of the potential issues for this poor performance discussed in the study are constrained lab time, large classes and insufficient feedback. In previous work, we found that automated feedback can help students to become independent learners when faced with new programming tasks (Rieser, PGCAP report, 2014). **The aim of this project was to extend this previous work to help first year CS students to become independent learners and master the art of programming during their first year at the university.**
4	***Project overview & aims:*** The main objective of this project is to develop a platform which automatically generates feedback for programming projects to **provide students with immediate formative feedback** and **help them to become independent learners**. Furthermore, this platform will provide students with a **uniform flexible learning experience across the different campuses.** The basis for the feedback platform is a technique known as *unit testing*, which enables us to write a set of tests that students can run their code on to look for mistakes. Our focus is on the Java language, which is used in our first year courses. Java supports unit testing via the JUnit framework. Using Junit, this projects builds an automated test harness for Java programming to generate formative feedback to the students. This type of automated feedback has been shown to be successful in, for example, teaching students how to code using Java in a different platform (Vihavainen et al. 2013). The tests are a piece of Java code, which are able to automatically find errors in the code and detect missing aspects. Here, **we study whether this automatic analysis can be used to generate meaningfull feedback**, for example, in terms of clues and tips how to improve the code; a technique called `*scaffolding'* (Vihavainen et al. 2013).

	We run two programming courses, Software Development 2 (SD2) and Software Development 3 (SD3), in parallel during second semester for first year Computer Science students. Before attending these courses, students will already have had some basic introduction to programming in Java via Software Development 1, which is taught in the first semester. They are expected to deepen their knowledge and skills in these two more advanced courses. A second objective is to develop a joint coursework for SD2 and SD3, which utilises the feedback framework. A key challenge is then to find which type of challenges that are amendable to automatic feedback, and which type will require help from lab helpers/lecturers.
5	***Activities and details of project steps taken to achieve aims:*** To achieve this goal, we the following approach is taken: 1. Develop a specification of the requirements for the unit tests based on the existing lab exercises. 2. Develop generic unit test cases, answering the question of *“what type of feedback can we provide automatically?”* 3. Develop joint coursework for SD2 and SD3 and adjust unit tests. 4. Test platform, debug, and review units with respect to their pedagogical quality regarding the feedback they produce (in collaboration with academic staff). 5. Disseminate activity through appropriate channels. For example, academic publishing, computing in schools initiative, etc.
6	***Key points including challenges your team may have encountered:*** Firstly, note that the project is ongoing and we other challenges are likely to occur during later stages of the project. Early experiments have indicated that unit testing is a suitable mechanism to provide feedback concerning shallow learning but proves to be difficult when applied to deep learning. Deep learning involves the critical analysis of new ideas, linking them to already known concepts and principles, and leads to understanding and long-term retention of concepts so that they can be used for problem solving in unfamiliar contexts. In contrast, surface learning is the tacit acceptance of information and memorization as isolated and unlinked facts. In terms of exercises for computer programming this means that unit tests are suitable to check whether students were able to solve a prescriptive task (shallow learning), but they are not suited to test whether the students is able to creatively apply the underlying principles to solve an unseen problem. We are currently investigating how to use interactive systems, such as tutorial dialogue (Litman, 2013).
7	***Describe specific project outputs so far:*** We have created a series of tests using the two libraries: Junit (described above) and Reflect (enables us to examine runtime features of programs). This enables to check for the existence of specific Java features such as classes, variables and methods. Crucially, it allows us to check whether students have successfully accomplished simple program creation tasks. When one of

	these tests fails messages are displayed to the student offering advice that is relevant to the failure point. For example if the test fails to detect a given class then a message is displayed offering information on the correct creation of classes, common pitfalls such as not closing brackets and the fact that class names are case sensitive. The tests are nested which allows us to display advice that is relevant to only the currently encountered problem. This allows us to not overwhelm the student with information. This provides immediate advice when it is needed to progress in a task; a feature of the educational *Scaffolding theory.* One of the drawbacks of these tests is that they look for specific class, method or variable names. This means that students performing the coursework need to be told specifically what to create, potentially limiting their opportunities for creativity and problem solving. This is not such a problem in the early stages of the course where the goal of the coursework is to simply teach the mechanical how-to of program writing but will become more of an issue as the course progresses and the tasks, concepts and learning outcomes become more complex (see above discussion on shallow vs deep learning). One solution we looked at was trying to identify whether classes or methods existed by their output rather than their name because it is the output, and not the name, that is the relevant part . For example, if the task is to create a method to sort strings then we give all methods a string and see if one of them returns the sorted string. However this too becomes problematic as tasks become more complex as every method would need every test applied to it. There is also the problem, once again, of restricting the way students solve a problem if the tests are looking for methods that act in a specific way. It is also difficult to identify when a student solution is nearly correct using this method. A missing semi-colon will carry the same weight as a fundamental error in the way a student is approaching a problem, which makes targeting advice difficult. One of the learning outcomes for the coursework is that students are aware of the concepts of *coupling and cohesion*. This has proven difficult to test for. One early idea was to the capture the stack trace of the program to try and see how many steps were being taken but this has not produced anything useful so far. Another methods is to count how many methods each class has and how many calls are made to other methods. These numbers could be compared against an "ideal solution" to see if they fall within an acceptable threshold. However the concept of an "ideal solution" may be at odds with student creativity. One of the goals is to make the testing framework generic so that it may be used on multiple pieces of coursework. The *QuickCheck* library allows randomised test data to be created but the type of randomised data will still have to be manually set up as it will be specific to each coursework task. So far randomised test data seems like it will be more useful in catching edge cases in student code rather than creating a blanket set of tests for all pieces of coursework.
8	***Please describe how your project has contributed to the to the Heriot-Watt University strategy and priorities for Enhancement:*** ***1. Shorter time to completion for Post-graduate Research students (PGR)*** ***2. Improved retention and progression of undergraduate students:***

	This project has contributed to improve the retention and progression of undergraduate students in the following ways: **Relevance** This project is aimed at students who just have entered their first year at the university, and transition from being a school pupil to become an independent learner. The proposed platform will enable students to learn at their own pace and engage with the material in a novel method making use of modern technology. In addition, it should enable students who are direct entrants to later years of relevant programmes to complete relevant sections prior to their arrival to help with their integration into the cohort. This project supports the school's L&T strategy across many objectives. Firstly, one of main goals within the L&T strategy is the integration of modern technology into our teaching methodology to improve the student experience. This project directly tackles this by developing a more flexible environment and new methods to engage with the material. Secondly, as the school moves to teaching more courses at different campuses and through partnerships with other FE and HE institutions this project will help provide students with a uniform learning experience through the use of modern technology. **Innovation** This project looks to use modern technology in an innovative way to foster engagement of the students with the material and provide a uniform teaching experience across various Heriot-Watt University campuses. The use of technology in the proposed project is an innovative teaching method.
9	***Describe how you are sharing good practice within Heriot-Watt and beyond (e.g. plans for papers, attendance at conferences):*** **Impact on teaching:** This project has multiple immediate impacts on improving student experience within the department. The initial test platform will be utilised in the first year of teaching programming within the department. Verena Rieser, Michael Lones and Gudmund Grov are currently in charge of this teaching within the department and will assist the integration of the development platform within the teaching syllabus. This will have impact for students across several of Heriot-Watt's campuses. The vast majority of courses in later years of the Computer Science programme (including Programming Languages, Data Structures and Algorithms, Web Programming, Computer Network Security, Artificial Intelligence, etc.) assume that students have acquired solid programming skills during their first year. Students that haven't done so, will struggle in their later years. Thus, providing an improved method of delivery of programming teaching will have a positive impact on these courses as well. Furthermore, this platform will be integrated in a joint coursework between Software Development 2

	and 3, which is being developed as part of this project. This will allow students to integrate their knowledge in a joint project. Having an automated method to assess and provide feedback to students, also means a reduction of additional workload for the lecturers, while students profit from having more practical experience. **<u>Dissemination:</u>** This research was disseminated at the HWU Learning & Teaching colloquim on 11th March and QAA workshop on Using Technology to Manage Assessment and Feedback. After analysising the result of the result of the joint coursework we plan to submit a paper to suitable CS eduction conference, such as ICER. Alternative, we may target a broader audience and submit an article to CACM.
10	***Next steps:*** As the project is ongoing, the next step is to complete the desired tasks. This will involve further developing the framework and developing the joint coursework. Still, this initial exploration of using an automatic testing harness to provide feedback on programming performance as opened interesting research questions which we will not be able to complete during the lifetime of the project but hope to explore in future work: • Firstly, we plan to evaluate the platform with students during Software Development courses and refine the platform based on student feedback. We may apply for further QAA funding to support this evaluation. • Secondly, we want to investigate how to support deep learning using automated feedback by, for example, using interactive tutorial dialogue technology (Litman, 2013). In order to achieve this, we plan to propose student projects, e.g. as part of their MSc or Honours dissertation, as well as applying for PhD funding to pursue this research further.
11	***Additional information:*** Please note that the project is ongoing is still is just an outline documents. A complete report will be submitted at the end of the project. ***<u>Bibliography:</u>*** *Michael McCracken, Vicki Almstrum, et al. 2001. A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. In Working group reports from ITiCSE on Innovation and technology in computer science education. ACM, New York, NY, USA, 125-180.* *Arto Vihavainen, Thomas Vikberg, Matti Luukkainen, and Martin Pärtel. 2013. Scaffolding students' learning using test my code. In Proceedings of the 18th ACM conference on Innovation and technology in computer science education (ITiCSE '13). ACM, New York, NY, USA, 117-122.*

	Diane Litman. Enhancing the Effectiveness of Spoken Dialogue for STEM Education. Proceedings Workshop on Speech and Language Technology in Education (SLaTE), pp. 13-14, Grenoble, France.