Using Web-based Peer Assessment in Fostering Deep Learning in Computer Programming

Jirarat Sitthiworachart ** and Mike Joy

Department of Computer Science, University of Warwick, Coventry, CV4 7AL, UK
jirarat@dcs.warwick.ac.uk,
WWW home page: http://www.dcs.warwick.ac.uk/

Abstract. Active learning is considered by many academics as an important and effective learning strategy. Students can improve the quality of their work by developing their higher cognitive skills through reflection on their own ideas and practice of analytic and evaluative skills. Peer assessment is one of the successful approaches which can be used to enhance this deep learning. In this paper we discuss a novel web-based peer assessment system to support computer programming courses. We discuss the educational rational for the system, and the deep learning theory, report on its deployment on large programming modules. The preliminary results indicate that the system has successfully helped students to develop their higher cognitive skills in learning computer programming.

1 Introduction

Assessment is a tool for learning, but traditional assessment methods often encourage surface learning, characterised by memorisation and comprehension of information. Deep learning, such as creating new ideas, and critical judgement of a student's work, can be encouraged by the use of peer assessment [1–3]. When students evaluate each others' work they think more deeply, see how others tackle problems, learn to criticise constructively, and display important cognitive skills such as critical thinking [4, 5]. As part of a study investigating the extent that peer assessment can promote deep learning in a programming course, we have developed a novel web-based peer assessment tool specifically designed to support computer programming courses, which automates the marking process, anonymises the participating students, and provides comprehensive monitoring facilities for the teacher.

This paper first gives an overview of surface and deep learning, which can be described by using six categories of learning in Bloom's Taxonomy [6], which require students to use thinking skills at each level, starting from the simplest to the complex. Deep learning approaches and the key elements to encourage a deep learning approach are discussed. Finally, we describe the novel web-based peer assessment system, and report on its deployment on a large computer programming course.

** Please consider for "Best Student Paper" award

2 Deep Learning

Learning is increasing knowledge and understanding. Some students see learning as a matter of memorising and reproducing knowledge in ways acceptable to the teacher and only to cope with course requirements. Others see learning as a way to establish personal meaning, by transforming the information and ideas in relation to their existing knowledge and experience [8]. Brown, Bull, and Pendlebury [9] define learning as "changes in knowledge, understanding, skills and attitudes brought about by experience and reflection upon that experience", and summarize that students are expected to have two main skills, which are knowledge and thinking skills (critical judgement).

2.1 What is Deep Learning?

Deep learning is characterised by students conceptualizing approaches, seeking interconnections between concepts and data, and engaging in reflection [10]. According to Biggs [11], deep learning is "the use of high-level, abstract cognitive processes that teachers want students to develop, which include explaining, arguing, reflecting, and applying knowledge to problems that are not in the textbook, relating new problems to established principles, and hypothesizing". Cox and Clark [12] describe deep learning as "the capacity to use explanatory concepts creatively, and leads to students' ability to think about problem situations and devise new solutions to those problems". Rosie [13] argues that deep learning is not a function or attribute of the learner but is a strategy that students can adopt. He suggests that deep learning requires higher order cognitive skills, meaningful engagement in and enjoyment of learning, and a desire to think conceptually rather than to amass detail.

Thus to achieve deep learning students need a base knowledge and comprehension of information, which is surface learning, and they need to be able to apply their knowledge to solve problems. Once they can do these things they can abstract from their knowledge and experience so that they can move beyond known solutions to new and innovative applications and to use judgement in evaluating alternative solutions [6, 12]. It can be concluded that students need to begin with surface learning and progress systematically to deeper learning approaches. For the purposes of this paper, we use Bloom's categorization of cognigative skills [6], (usually referred to as "Bloom's Taxonomy"), although others are available, such as Anderson and Krathwohl's revision of the Taxonomy [7].

2.2 Surface and Deep Learning

In order to develop students' intellectual abilities, Benjamin Bloom created a taxonomy which categorizes six levels of learning that advocate the development of intellectual abilities [6]. The categories in Bloom's Taxonomy can be divided into two levels of learning: surface learning and deep learning. Surface learning consists of the first three levels of Bloom's Taxonomy, namely Knowledge, Comprehension, and Application, which emphasise recall and application of trivial procedural knowledge. The final three levels of Bloom's Taxonomy, Analysis, Synthesis, and Evaluation, are combined into a problem-solving skills category or deep learning level [12]. It should be noted that each succeeding level assumes competence at an earlier level. For example, students should have knowledge and comprehension of information before they can apply their knowledge to solve problems. A brief description of the six categories of learning in Bloom's Taxonomy is given in Table 1.

Table 1. Six categories of learning in Bloom's Taxonomy

Surface Learning	1 Knowledge	remember and recall information
	2 Comprehension	understand the meaning of material by describing or reviewing material in one's own words
	3 Application	apply material in new situations to solve problems
Deep Learning	4 Analysis	break down material into its components by identifying parts and analyse their relationships
	5 Synthesis	combine parts together to form a new whole
	6 Evaluation	judge the value of material for a given purpose or make a decision based on appropriate criteria

The difference between surface and deep learning is relevant in analysing student learning intentions, learning styles, learning approaches adopted and learning outcomes [14]. In surface learning, students learn by passively re-producing [8]. They learn simply to memorise facts, study without reflecting, and accept the information given without question [8, 14]. In contrast, in deep learning, students learn by active transformation [8]. They seek to understand the issues and interact critically with the contents, to relate ideas to previous knowledge and experience, to examine the logic of the arguments and relate the evidence presented to the conclusions, and become actively interested in the course content [8, 14].

Therefore memorisation and comprehension of information are the strategies in surface learning, but creating new ideas and critical judgement of the contents are strategies in deep learning. In deep learning, students intend to understand ideas for themselves by relating those ideas to previous knowledge and experience [6, 8]. In contrast, students engaging in surface learning intend to remember facts and study without reflecting on either purpose or strategy [8]. Thus students should be encouraged to move from surface to deep learning by providing teaching and learning activities that support deep learning.

2.3 Deep Learning Approaches

There are many authors who have suggested a variety of approaches to help students achieve deep learning. For example, Entwistle [8] suggests that deep learning can be promoted through curriculum design, teaching, and assessment. Grauerholz [15] proposes teaching holistically to help students achieve deep learning. Nine strategies for fostering a deep learning approach has been published in the AAHE Bulletin [1], and are summarised in table 2.

Table 2. Nine strategies for fostering deep learning

Strategy	Method
Encourage independent learning	learning contracts, self-assessment, and peer-assessment
Supporting personal development	intensive group work
Presenting problems	using 'real world' problems out of which learning and action arise, the integration of knowledge from different disciplines and interaction
Encouraging reflection	encourage reflection include the use of learning diaries, reflective journals, and use of video and observers when learning skills
Using independent group work	emphasize independent group work include group-based project work and peer tutoring, in which students teach one another
Learning by doing	using games, simulations and role plays; visits; and work experience
Developing learning skills	combination of all the above
Setting projects	involves the application of knowledge to new situations, learner activity, and demands a high level of motivation
Fine tuning	through the introduction of active learning task and peer-group discussion into otherwise passive lecture classes

It can be concluded that perhaps the most important for encouraging a deep learning approach is reflective learning [1]. Moreover, setting projects — which involves the application of knowledge to new situations — is the most common strategy used for fostering a deep learning approach in higher education [1]. However, we have chosen to investigate peer assessment, since this is an approach which, although often used in the context of essays, has seldom been applied to computer programming courses. The skill of writing good software includes understanding of different approaches to the task, and stylistic and related considerations — these can be developed by evaluation of other programmers' solutions. Peer assessment also relates to the four key elements identified by Biggs as required for fostering a deep learning approach: the motivational context, active learning, interaction with others, and a well-structured knowledge base [11]. We have investigated how the peer assessment process can be adapted to work in the context of computer programming, and the effectiveness of that process.

However, deep learning is difficult to assess or see [15]. Some students may be resistant to deep learning, because it is hard work and students must be highly motivated to do it. Therefore, students should be provided with the tools to perform this higher cognitive activity and be stimulated to do it. For this reason we have chosen a web-based peer assessment tool to support our investigation.

3 Peer Assessment

Falchikov [16] defines peer assessment as "the process whereby groups rate their peers". Somervell states that peer assessment engages students in making judgements on the other students' work [17]. In the peer assessment process, students are involved both in the learning and in the assessment process. Peer assessment is primarily a tool for learning rather than for summative assessment [9]. Dochy and McDowell remark that "peer assessment is not only a tool to provide a peer with constructive feedback which is understood by the peer. Above all, peer assessment is a tool for the learner himself." [18]

Receiving many and frequent peer feedbacks can prevent some errors and provide hints for making progress in learning [19]. Thus the peer assessment process provides many benefits to students, including the following:

- it encourages of students' deep learning skills in programming by making judgements and providing feedback on other student's work [2, 3];
- students have opportunities to compare and discuss about what constituted a good or bad piece of work, which help them to improve their programming style and think more deeply about the quality of work [20];
- when marking, students realise mistakes that they had made in their own answers - the more marking students did, the better their own results became [3];
- students develop self-assessment and reflective learning skills [2, 17]; and
- students' understanding of the assessment process deepens [9].

3.1 Research Method

The peer assessment experiment was performed on 215 first year undergraduate students (189 male and 26 female) of whom 153 students' first language is English and 62 students who are not native English speakers. The age range of the students was 18-26, with most in the 18-20 band. The UNIX programming module in the authors' Computer Science department was chosen for this investigation. This module aims to give students a basic understanding of the UNIX operating system, and competence in programming using a UNIX shell.

Students learn how to design and develop programs in the shell, which is a programming language that allows programs to be written in many styles. There are three programming assignments in this module, which students submit via the department's "BOSS" online submission system [21]. The second of the three assignments was marked using a peer assessment process. The purposes of performing the experiment in peer assessment were:

- to investigate the extent that peer assessment in a programming course promotes deep learning;
- to assess the accuracy of students' judgements during a peer assessment exercise; and
- to provide evidence that peer assessment in computer programming has a positive pedagogical effect.

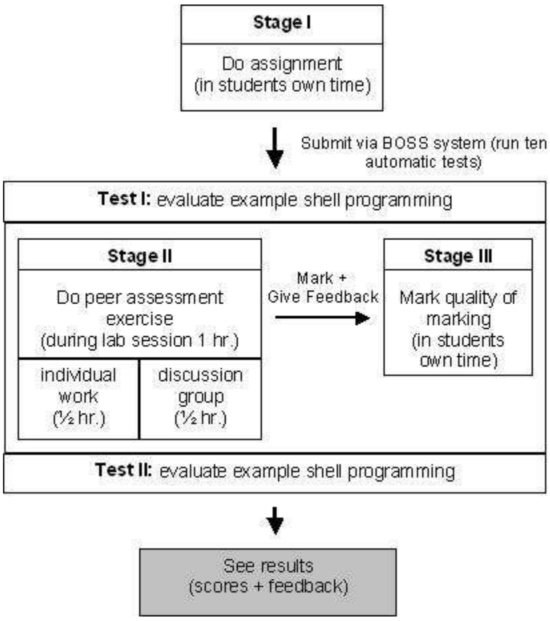

Stage I
Do assignment (in students own time)
Submit via BOSS system (run ten automatic tests)
Test I: evaluate example shell programming
Stage II
Do peer assessment exercise (during lab session 1 hr.)
individual work (½ hr.)
discussion group (½ hr.)
Mark + Give Feedback
Stage III
Mark quality of marking (in students own time)
Test II: evaluate example shell programming
See results (scores + feedback)

Fig. 1. Peer assessment process

Process This peer assessment exercise was divided into three separate stages, as shown in Figure 1. Test I and test II were provided in order to measure the students' evaluation skills, before and after the peer assessment. Students analyse and evaluate short example shell programs in test I and test II, which are similar in content but cosmetically different.

Stage I: Students do the assignment in their own time. Then they submit the assignment via the online submission system. Ten automatic tests are then run on the submitted programs.

Stage II: Students are divided into small groups (three students per group), each group consisting of students with a range of abilities. Each student is assigned three other students' assignments to mark during the first half hour of a lab session. Then they discuss their marking with the other students in their group, who marked the same assignments.

Stage III: In their own time, each student marks the quality of three markers' marking. This additional stage aims to develop students' critical judgement and encourage them to take the assessor role more seriously during stage II.

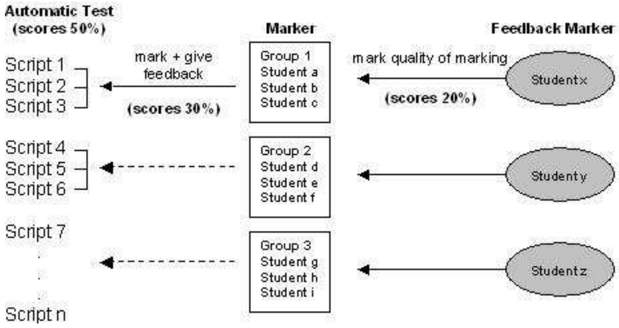


Fig. 2. Peer assessment mark scheme

Automatic test: The online submission system tests a student's assignment against different inputs to check whether it functions correctly. Ten tests are used.

Marker: Student markers mark assignments.

Feedback marker: Student feedback marker reports on the quality of the marking given by the three markers.

Script: Assignment that students submit via the online submission system.

Mark Scheme The marking scheme is illustrated in Figure 2. In this peer assessment process, 50% of the marks are awarded by the teacher (automatic tests) and the remaining 50% are awarded by the students (peer assessment):

Automatic Test		50%
Peer Assessment		
- Part I:	mark assignment	30%
- Part II:	mark quality of marking	20%

Peer marks are based on three markers; the average of the three marks is calculated. If one of markers does not appear to have marked work seriously (that is, does not appear to have read the program carefully and answer the marking criteria properly), the mark he or she gives will not be included in the average and the other marks will be scaled. The marking of assignments by students is possible since they are given guidance, automatic test scores and results, a marking scheme, and well-explained marking criteria.

3.2 Web-based Peer Assessment

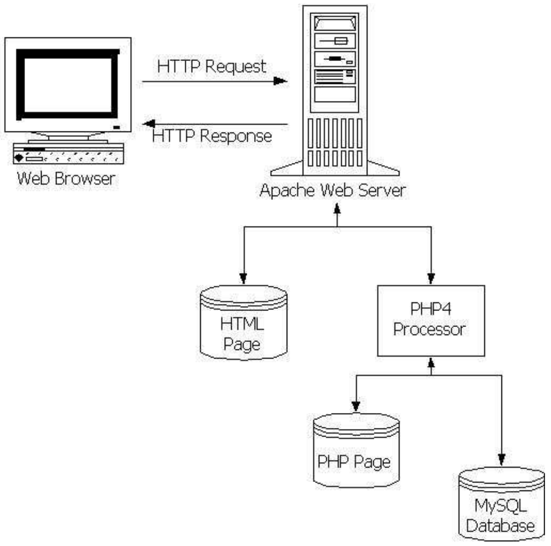
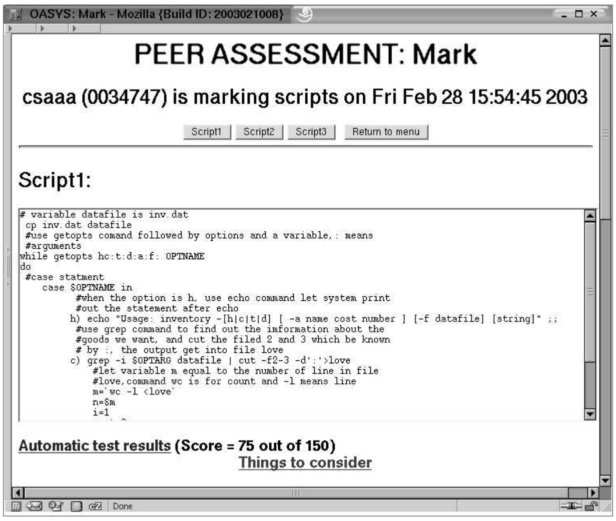


Fig. 3. Architecture of the web-based peer assessment system

The web-based peer assessment software uses the standard combination of Apache web server, the PHP4 programming language, and a MySQL database running on a Linux platform. This architecture is illustrated in Figure 3. Dynamic web pages are written in PHP4 and static web pages are written in HTML.

This web-based peer assessment provides anonymity for all users. Students are allowed to revise the marks they give until the marking deadline is reached. They receive a username and password by email before starting the peer assessment exercise. After students login, the menu page displays three steps for students to follow (i.e. mark assignment, mark quality of marking, and see mark). They can see the scripts that they have been assigned to mark easily by clicking on the script buttons (Figure 4). They can view the automatic test results by clicking on the link on each script page to open a popup window displaying the results. A "Things to consider" link is provided below containing marking guidance.



PEER ASSESSMENT: Mark

csaaa (0034747) is marking scripts on Fri Feb 28 15:54:45 2003

Script1:

```
# variable datafile is inv.dat
 cp inv.dat datafile
 #use getopts comand followed by options and a variable,: means
 #arguments
while getopts hc:t:d:a:f: OPTNAME
do
 #case statment
    case $OPTNAME in
          #when the option is h, use echo command let system print
          #out the statement after echo
         h) echo "Usage: inventory -[h|c|t|d] [ -a name cost number ] [-f datafile] [string]" ;;
          #use grep command to find out the information about the
          #goods we want, and cut the filed 2 and 3 which be known
          # by :, the output get into file love
         c) grep -i $OPTARG datafile | cut -f2-3 -d':'>love
             #let variable m equal to the number of line in file
             #love,command wc is for count and -l means line
             m=`wc -l <love`
             n=$m
             i=1
```

Automatic test results (Score = 75 out of 150)
Things to consider

Fig. 4. Assignment script on *Mark* web page

Mark Assignment In this visual inspection step, students mark and provide feedback on other students' assignments by answering nine questions about:

- Readability (comments, indentation, variable names);
- Correctness (correct output, appropriate error handling, correct exit status); and
- Style (easy to follow, well structured, use of appropriate utilities).

These are answered for each script by selecting simple multiple choices, i.e. *No*, *Partial*, and *Yes*. The default answer is set as *unmarked*. Students give a comment for each group of three questions. An explanation of the marking criteria is provided for each group of questions by clicking on the links on the left.

Mark Quality of Marking In this step, students mark the quality of marking given by each of the three markers on a particular script. They need to answer three questions about whetherthe suggestions the markers gave in each section (readability, correctness, and style) are relevant, well explained and useful to students. The marking given by the three markers is displayed at the top of the page and the student enters the feedback marks at the bottom.

See Mark In this final step, students can see their mark from both the automatic test and the peer assessment. A *Marking calculations* link at the bottom of the page provides an explanation of how the overall mark is calculated. If the students do not mark any of three scripts, they may lose some marks. The full mark and comments that the three peer markers gave the student's assignment are also available. This also includes the full mark that they were given based on the quality of their own marking.

Monitor Marking The monitor marking web page reports the students' marks and any absent markers, and is only available for tutors. The highlighted columns show the standard deviation of the three markers for both Step I and Step II in order to know how spread out the marks are. If the standard deviation is less than a preset value, it is deemedacceptable, but if the standard deviation is more than a given upper limit, it means the marks from the three markers have a very wide range, and the tutor may have to reconsider the marks for that student. The tutor can access each script by using the "Script ID" box at the top of the web page.

4 Results

Students' opinions At the end of the process, each student was required to fill in a detailed online questionnaire, and the following preliminary observations suggest that the exercise has been beneficial.

- 69% of students realise mistakes that they made in their own answer when marking other students' work.
- 76% of students discuss with their groups when marking and think this discussion helps them understand more about the assignment.A few students find starting a discussion difficult.
- 58% of students feel comfortable when assigning marks. A few students did not fully understand the marking criteria.
- 65% of students are satisfied with their mark from the peer assessment, and considered that the peer feedback they received was relevant and useful.

- 80% of students agree that seeing good and bad programs help them in learning programming, and marking helps them to think more deeply about their own work.

For most of the questions in the questionnaire, the responses when compared by gender were broadly similar. However, male students appeared to feel significantly more comfortable assigning marks than did the female students. Almost 80% of non-UK students would like to recommend the peer assessment to friends as a way of learning more, but only 50% of UK students would like to do this, and this may form the basis for further investigation.

4.1 Evaluating students' evaluation skills

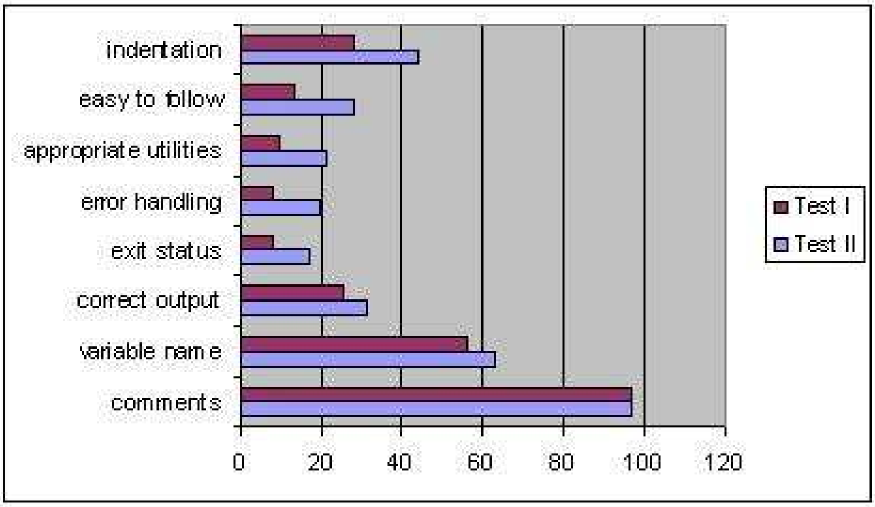


Fig. 5. Pre and post peer-assessment test results

In an attempt to obtain a controlled measurement of the effect of peer assessment, the students were asked to complete two tests additional to the assessment process discussed above. Test I, run before the peer assessment exercise asked students to analyse and evaluate a short shell program. Test II was very similar in content but had cosmetic differences in order that its content would not be recognised immediately as being essentially the same. The peer assessment exercise (including both tests) was completed within one week, during which no tuition relating to the course was delivered. The numbers of times that students commented on various (unprompted) aspects of the code were counted. A summary of the results is displayed in Figure 5, which suggests that when evaluating a shell program, the students were able to characterise more finely after they had been through the peer assessment process.

4.2 Quality of students' marking

In order to ascertain the accuracy of the students' marking, the programs were also marked independently by an experienced tutor, and each student's mark was then compared against the tutor mark. The marks awarded by the students were mostly higher than those awarded by the tutor, the means differing by approximately 18%, and scaling of peers' marks yielded results which, in almost all cases, matched the tutor's. This is consistent with other studies which in which peer marks are consistently higher than tutors [22]. With guidance, and design of appropriate information flow, peer assessment can be of a similar quality to that made by subject experts.

5 Discussions

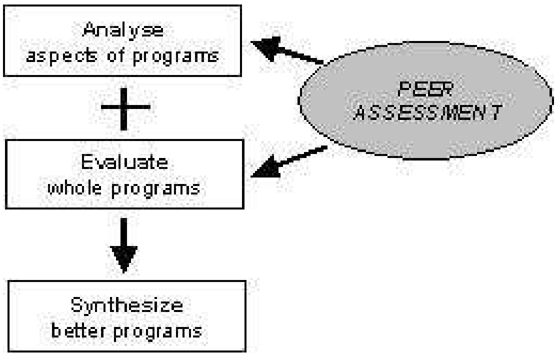


Fig. 6. Deep learning approach

The results of our investigation indicate that peer assessment encourages students to develop higher level cognitive skills in learning computer programming. This includes the analysis and evaluation of programs, which leads students to develop new creative approaches and styles when writing programs (as illustrated in Figure 6):

- Analyse aspects of programs - peer assessment starts students thinking about what aspects of their programs are wrong and what works well when they compare their work with other students' work.
- Evaluate whole programs - making judgement on other students' work helps students to evaluate their own programs. They can recognise what constitutes good and poor programs.
- Synthesize better programs - students analyse and evaluate the different styles of solving programming problems, which help them to create better programs.

The students' responses generally appear to support our view that students can learn from each other through this process. Seeing different ways of solving programming problems and marking each other's works helps students' self-assessment and writing better programs:

> "I got the chance to observe two scripts that used different methods than my own solution to satisfy the specification. In order to be confident in my ability to mark these scripts fairly, I had to spend a long time studying them and hence acquiring an improved knowledge of how shell scripts are composed."

> "Marking others' work helps me criticise my own work and remind me of my own problems."

Most students seemed satisfied with their marks, and considered that with adequate guidance the marks from peers could be as reliable as the marks from a tutor. However, some students think the marks awarded by students are graded using different standards to those awarded by tutors. Markers could only base marks on how good the script was compared to their own answers, and some students therefore think no student is really qualified to mark another student's work, as they are not trained for marking and are not experts:

> "I would say peer assessment in a better way to learn how to write a good program. Nevertheless, some markers may not have the skill of marking and understanding of script."

> "I think it is hard to mark a student when you've never marked assignments before, especially UNIX scripts as I never had any previous experience with it. Additionally, the things I consider good or bad may not be the same for other people."

It was difficult in the students' view to avoid friendship marking, resulting in over-marking (they often felt favourable towards their friends) [20]:

> "When marking I was inclined to be generous because I expected that everyone else would be to me."

> "Very hard to do when you are marking someone who is technically in the same boat as you."

6 Conclusions

We have described a deep learning theory and peer assessment process, together with supporting web-based software, which we have used to test the effectiveness of peer assessment in learning programming languages. The process we have used is novel, since students are engaged not only in marking each other's work, but also in evaluating the quality of marking of their peers. Students use deep (analysis, evaluation and synthesis) rather than surface learning strategies in the peer assessment process, contributes to effective learning. Students also realise their own strengths and weaknesses when marking a good or poor piece of work. Preliminary evaluation of the exercise indicates that it has contributed positively to the students' learning experience.

References

1. AAHE Bulletin: Deep Learning, Surface Learning. AAHE Bulletin **45**(8) (1993) 10–13
2. Topping, K.: Peer assessment between students in colleges and universities. Review of Educational Research **68** (1998) 249–276
3. Bhalerao, A., Ward, A.: Towards electronically assisted peer assessment: a case study. Association for Learning Technology (ALT-J) **9**(1) (2001) 26–37
4. Sluijsmans, D., Dochy, F., Moerkerke, G.: Creating a learning environment by using self- peer- and co-assessment. Learning Environments Research **1** (1999) 293–319
5. Tsai, C., Lin, S., Yuan, S.: Developing science activities through a net worked peer assessment system. Computers and Education **38** (2002) 241–252
6. Bloom, B.S. (Ed.): Taxonomy of educational objectives: The classification of educational goals: Handbook I, cognitive domain. New York, Longmans (1956)
7. Anderson, L.W., Krathwohl, D.R.: A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. New York, Addison Wesley (2001)
8. Entwistle, N.: Promoting Deep Learning Through Teaching and Assessment. In: Assessment To Promote Deep Learning. American Association for Higher Education (AAHE), Washington, DC (2001) 16–26
9. Brown, G., Bull, J., Pendlebury, M. (eds.): Assessing student learning in higher education, Routledge, London (1997) 21–39
10. Jenkins, T.: The Motivation of Students of Programming. Proceedings ITiCSE 2001. ACM SIGCSE **33**(3) (2001) 53–56
11. Biggs, J.: Assessing for Quality in Learning. In: Assessment To Promote Deep Learning. American Association for Higher Education (AAHE), Washington, DC (2001) 70–73
12. Cox, K., Clark, D.: The Use of Formative Quizzes for Deep Learning. Computers and Education **30**(3/4) (1998) 157–167
13. Rosie, A.: Deep Learning: A dialectical approach drawing on tutor-led. Learning in Higher Education **1**(1) (2000) 45–59
14. Beattie, V., Collins, B., McInnes, B.: Deep and surface learning: a simple or simplistic dichotomy? Accounting Education **6**(1) (1997) 1-12
15. Grauerholz, L.: Teaching Holistically to Achieve Deep Learning. College Teaching **49**(2) (2001) 44–50
16. Falchikov, N.: Learning Together: Peer tutoring in higher education. Routledge Falmer, London (2001)
17. Somervell, H.: Issues in assessment, enterprise and higher education: the case for self-, peer and collaborative assessment. Assessment and Evaluation in Higher Education **18** (1993) 221–233
18. Dochy, F., McDowell, L.: Assessments as a tool for learning. Studies in Educational Evaluation **23**(4) (1997) 279–298.

19. Lin, S., Liu, E., Yuan, S.: Web-based peer assessment: feedback for students with various thinking-styles. Learning **17** (2001) 420–432
20. Brindley, C., Scoffield, S.: Peer Assessment in Undergraduate Programmes. Teaching in Higher Education **3**(1) (1998) 79–89
21. Luck, M., Joy, M.: A Secure On-line Submission System. Software - Practice and Experience **29**(8) (1999) 721–740
22. Magin, D., Helmore, P.: Peer and Teacher Assessments of Oral Presentation Skills: how reliable are they? Studies in Higher Education **26**(3) (2001) 287–298