TOPS – Collaboration and Competition to Stretch our Most Able Programming Novices

Su White [1], Janet Carter [2], Stephan Jamieson. [3], Nick Efford [4], and Tony Jenkins[5]

Abstract -

Finding effective means of meeting students' varied needs during introductory programming classes is a perpetual challenge. Maintaining motivation and a sufficient level of engagement across an undergraduate group with diverse prior experience is not a simple task. Claims for successful approaches include forms of differentiated teaching and paired programming. Competitions run by professional bodies and the software industry are often used to provide further external motivation. This paper presents the outcomes of a collaborative initiative across four universities which drew on aspects of both these approaches. Academics in the partner institutions had already implemented specific practice to accommodate the variety of student needs. The TOPS project was designed to involve and extend students through the processes of devising and competing in an inter-university challenge. Analysis of the outcomes has enabled the development of further understanding and good practice in this important area.

Index Terms – active learning, computer science education, paired programming, student motivation.

INTRODUCTION

It is unusual to find any first year programming class solely composed of novices. Typically students in a cohort possess a range of prior experience, which generates challenges when determining the most effective way in which to manage and teach the class. Commonly the course aims to ensure that all students receive a sound introduction to the chosen language and appropriate software engineering principles. Specific issues in teaching to these aims include how to ensure the maximum benefit and engagement for each of the participants irrespective of their skill level at the beginning of the course. A key objective which emerges, therefore, is to maintain motivation throughout the range from absolute beginner to the most self-confident or experienced.

The TOPS project was established as a collaborative initiative between four UK universities. The project was begun towards the end of 2006 and designed to be run and evaluated within the same academic year.

Academics met initially to plan the broad structure of an inter-university challenge. They compared and considered the curriculum for introductory programming at each institution and the broad constraints of possible programming challenges were agreed. It was agreed to set challenges in Java. Students would work in whatever environment they were accustomed to using.

Each university was to provide no more than five paired teams to compete under time constrained conditions at a central venue on the competition day. Programming challenges would be devised by whole university teams, but undertaken by programming pairs (up to five per university). It was anticipated that the activities might appeal to, and be especially appropriate for students with pre-existing programming experience. However no student would be precluded from participating.

Student teams at each university were recruited and asked to devise a challenge to be undertaken by their fellow competitors. Each task was to be designed to last one hour and to be undertaken by programming pairs. Each team also provided a mark scheme to be used to guide marking.

This introduction section has provided a basic explanation of the project. The background section of the paper will outline the educational context and the educational objectives embedded within the original project specification. The Methods section will provide an account of the how the project was actually realized. The discussion section will examine the outcomes of the project and present our evaluation of the project activities. The conclusion section will highlight key findings and suggests future work.

BACKGROUND

The best way to teach introductory programming and what makes a good programmer is a perennial discussion point amongst computer science educators. The literature tracks the emergence of new methods and current thinking. When considering the nuances of introductory programming courses there have been many insightful and constructive responses.

Approaches which have been designed to gain insight into this crucial area of undergraduate learning include Authentic activities, paired programming, differentiated teaching, and creating disciplinary commons to enable sharing of

[1] Su White, University of Southampton, saw@ecs.soton.ac.uk
[2] Janet Carter, University of Kent, J.E.Carter@kent.ac.uk
[3] Stephan Jamieson., Durham University, stephan.jamieson@durham.ac.uk
[4] Nick Efford, Leeds University, nde@comp.leeds.ac.uk
[5] Tony Jenkins, Leeds University, tony@comp.leeds.ac.uk

1-4244-1084-3/07/$25.00 ©2007 IEEE

academics' and instructors' experiences and understandings [1-7].

Project Philosophy

The TOPS project was designed to incorporate activities of benefit to academics and participating undergraduates. The activities undertaken were vehicles for achieving understandings not normally achieved through the usual programming classes; by academics organizing the competition activities and by the students participating in the competition. Before outlining the activities and their broader objectives in some detail, key objectives are presented in Table 1 below.

TABLE 1
PURPOSE OF TOPS PROJECT ACTIVITIES

Academics	Students
Demonstrate proof of concept	Extend and motivate programming activities
Share current practice	Gain insight into the curriculum
Peer observe across universities	Undertake authentic time-constrained paired-programming

Proof of Concept

Although initially the competition was a collaboration between four broadly similar universities, it was designed to be a proof of concept for more widespread use within an educational community which shares broad approaches to the teaching of computing and computer science as a first degree.

Sharing Practice

The structure of the activities sponsored by the project provided a device for sharing good practice between academics. Participating academics needed to familiarize themselves with the detail of the curriculum from their partner institutions, discussing details of delivery and assessment methods built awareness of good practice.

Peer Observations

Peer observations were designed to create a sense of place, and also to enable comparisons of student cohorts at the time of the observation. For the academics it would provide additional time to prompt reflection and internal comparison of the approach they adopted at their home institution.

Extending Motivation

The competition and the collaborative task of setting the challenges were devised to be a means of extending the more able students in programming activities which are normally outside the curriculum. The tasks were specifically constrained to complement the existing teaching syllabus and classes.

Gaining Insight

Although independent competitions are often used to motivate students, the project team believed that activities which were designed in the specific context of existing curriculum could have greater educational strengths by virtue of their purposefully designed relevance. It was considered that students would gain additional insight into the purpose of the syllabus and the curriculum. They would also gain a deeper awareness the functional strengths of various programming constructs as they were engaged in the collaborative task of setting a challenge to their fellow competitors.

Authentic Tasks

Authenticity of task was introduced through participating in the competition; working as pairs to complete the challenges under time constrained conditions. Their work would be presented to fellow competitors and judges with explanations of the process and decision making undertaken. The activity and judging process was designed to stimulate student motivation. The message embedded in the structure of the activity and the judging process would clearly demonstrate the value of integrating professional and technical skills.

METHOD

The competition objectives have been outlined in the background section. However when moving from planning to implementation some small changes were made to the way in which the project activities took place. The rest of this section outlines how the process was realized in chronological order. Implications of changes between plan and realization will be considered in the discussion section below.

Planning – the academic perspective

There were two meetings of academics from all of the participating universities before the contest was staged. The agendas for the meetings were a mixture of operational detail and educational planning. Academics on the project met and discussed possible challenges.

Timings were constrained by a range of factors including different teaching term lengths, individual teaching and academic commitments, and travel times and budgets. Discussion was needed to determine differences between circumstances and teaching practices at the partner institutions. For observation purposes institutions were paired on the basis of geographical proximity. Further paired meetings then took place.

At the four partner meetings details of the curriculum, the syllabus and the typical teaching process were shared and discussed. The academics faced three objective tasks which related to the student competition. To ensure that:

- The student teams would construct viable and realistic challenges for the programming pairs;
- The challenges were appropriate for the level of expertise and prior learning;
- The challenges could realistically be attempted and potentially be completed within one hour;

Without any limiting constraints the academics found it difficult to visualize what types of challenges might be set by the students. It was agreed that providing a scenario for the challenge would assist the students in their discussions, planning and specification of the pair programming challenge.

Sponsorship had been found which enabled the contest to be hosted at a software industry technical conference. It was

therefore suggested that the scenario offered to the students would specify the challenge within the context of an activity appropriate or useful to students attending such an event. Some aspect of creating social networking software would fit within this bound. It was agreed that tasks should be appropriate to the learning objectives typically covered in introductory programming classes.

Planning – the student perspective

Academics recruited students to participate after the first four site academic meeting. Students were made aware of the competition via announcements in scheduled lectures, and blanket email calls for participation. For reasons of equity the competition was made open to any students in the cohort, although the majority who put themselves forward had programming experience prior to University. Although the travel budget limited attending competitors to three pairs per site, some reserve students were recruited who participated in the planning stages of the activity.

Students had approximately three weeks to specify their challenges. Student teams were set interim deadlines for the first draft of their challenge. Academics at the respective sites acted as mentors on the task, but left all detail of the challenge specification to the student team. In order to provide external motivation for the challenge setting, a team prize was offered for the best constructed challenge. A prize was also being offered for the best programming pair.

The students produced a range of challenges which fitted the scenario to varying degrees. Most teams constructed trial code to test the viability of their task. Classic behaviors within the teams emerged which is typically associated with more formal group project activities. Team outputs appeared to be influenced by strong personalities – often those individuals who possessed the strongest technical skills. Student teams appeared to enjoy the task of setting the challenge. All teams spent a large amount of time developing their task, and wanted to work right up to the final deadline refining their task.

The contest – the student perspective

Three of the university teams arrived at the opening of the conference on the morning of the contest. One team had stayed overnight – due to the long distance they had to travel to the venue. The three remaining teams all had early morning starts. For some students this resulted in the authentic activity of dealing with work life balance – juggling conflicting demands of their social life and university commitments. The result was little or no sleep, but they did not seem troubled by this issue.

All of the academics had worried that an early start might result in depleted teams, but that concern was unwarranted. A full complement of teams attended the event. For the majority of students it was the first industry conference/exhibition event they had attended. Typically they were impressed by both the technology and the food.

It was suggested that the best way to work would be with one laptop between two, although in some cases individuals or university teams had come equipped with additional technology. No effort was made to constrain the students from using this equipment; academics did not believe the students would gain any advantage from using additional technology.

University teams had brought multiple paper copies of their challenge with the mark scheme. They also had electronic copies of any required code or other resources.

The atmosphere during the first challenge was intense. The activity had started later than the planned time due to last minute changes in the conference schedule beyond the control of the project team. Working arrangements were ad hoc and it took a little time for the pairs to settle. The advantage of this room layout was that it in no way resembled an examination room.

Challenges were started and stopped using a whistle. Academics from each institution then collected their students efforts on a memory stick to hand over to the markers. After a fifteen minute break the next challenge began. Students undertook two challenges before lunch and one challenge after lunch. By the second challenge students pairs seemed to have settled into the task and the atmosphere became noticeably calmer.

Figure 1 below show the pair who would ultimately win, working together on one of their tasks.

FIGURE 1
STUDENT COMPETITORS DISCUSS ONE OF THE CHALLENGES.

The contest – the academic perspective

The day of the contest had been carefully scheduled as is shown in table 2 below. All the academics agreed that they had found the time leading up to the challenge personally stressful. In most cases they had felt anxiety surrounding their students' performance relative to those from other universities. This anxiety may have been heightened by the need to lead a group of students to arrive at the venue on time for an early morning start.

The academics were also responsible for distributing their teams' challenges and marking schemes, setting up the working layout of the room, and mentally rehearsing the prepared time schedule to reassure themselves that things would be able to go to plan. They were able to complete these immediate pre-contest tasks whilst their students attended the opening keynote, apart from the late arriving team, who were at that stage grappling with public transport.

TABLE 2
COMPETITION SCHEDULE

Registration	Three university teams
Coffee, visit stands	
Keynote	Academics set up room
Final team arrives	
First Challenge	
Break	
Second Challenge	Academics begin marking
Lunch Break	
Third Challenge	
Tea Break	Academics complete marking
	Judges consolidate marks
Prize Giving	

When the students began the challenges the academics were initially able to relax. However once the first challenge had been completed academics were faced with the task of marking the work immediately.

Tasks were distributed across the team which was augmented by two external judges from the funding agency and the sponsoring host company. Figure 2 below shows one of the judges working on a marking scheme early in the proceedings. It soon became clear that the judging tasks were uneven. Happenstance recruitment of a known and suitably experienced academic who came across the judges and academic team over lunch enabled the judging to be completed in close to the original planned timeframe.

All agreed the marking was daunting, although using the student teams' marking schemes enabled them to reach clear and confident conclusions when determining the winning programming pair. The judges canvassed the opinions of participants when judging the challenges and evaluating their quality, and reached a unanimous conclusion in awarding the prize for the best challenge.

FIGURE 2
ONE OF THE JUDGES REVIEWS A MARKING GRID.

DISCUSSION

The project was run as a proof of concept. The competition was a second semester activity for first year students. The activity supplemented existing differentiated approaches which seek to stretch the most able students. Four sets of three teams of programming pairs took part in the contest on the day of the competition. Two further reserve students attended. A further two participated in the challenge setting but did not attend. The numbers participating are not sufficiently large to perform meaningful numerical analysis, although the experience of the event can be used to evaluate whether it would be plausible to extent this style of activity to larger numbers of competitors (universities and programming pairs) in the future. Activity of this kind is dependent on finding adequate sources of funding or sponsorship.

The project set out to stretch our most able programming novices incorporating activities which demand integrating knowledge skills and understanding beyond the demands of the normal curriculum objectives. However for reasons of equity, participation in the activities was made available to any of the first year undergraduate students who were studying relevant programming modules. The selection process thus precluded all most able novice students from each university participating, although the majority of students who did participate fitted that criterion. The tasks they engaged with were more stretching than normally found in their programming assignments; the tasks incorporated higher level (integrative) objectives. Students' qualitative evaluations indicated that the tasks met the original criteria.

The project was run on a very tight budget. It was felt that a limited number of face to face meetings for the collaborating academics were important. Meeting facilitated the widest possible discussion, and assisted in understanding similarities and differences between the practice at the different universities. Unfortunately travel costs had increased since the original specification. This limited the number of participating students to six per institution. Sponsorship provided a venue and prizes. Originally it had been hoped that all students could have met together prior to the competition, the increased cost of travel meant this option was only available for one university, the remainder of the students traveled to the central location on the morning of the contest. Tight schedules and tight budgets resulted in a stressful experience for the academics involved, although all agreed that it had been a useful and worthwhile process, and were prepared to seek additional future funding with a view to extending the activity to additional institutions.

Student comments were overwhelmingly positive e.g.:

- *"Working together was great, everyone worked amazingly well in teams…I felt I learnt an enormous amount from the activity, and I thank you sincerely for making it available to us".*
- *"I liked that we were supposed to work at our natural pace and that we had to think".*
- *"It was really intense, but great fun".*
- *"Its great – thank you for organizing it".*

Scheduling the event was bound to have some problems. Different universities have different teaching schedules, and students will necessarily have to balance this type of additional activity with existing workloads which they may

see as more urgent and important. Some students suggested that staging the event in custom university labs would have been preferable, although the feeling of the academic team was that staging the contest as part of a commercial software industry event gave an appropriate out of university tone to the proceedings.

There was clearly a need to deal with the heavy task of marking. Discussion on the day of the event focused on perhaps using small teams of post graduate students as marking teams. Coupled with tighter specification of the tasks, and greater uniformity across the marking schemes it might be possible to increase the number of institutions and student teams who were participating. Reducing the number of participating teams from each institution could also be used as a device to extend the number of participating universities, although the numbers who took part already represent less than ten percent of an average 80 student computer science/computing cohort.

Another way of reducing the marking load would be to extend the challenges. Resulting in fewer, longer challenges, e.g. a two-hour challenge in the morning, another in the afternoon. Variation in the task might also be achieved by spreading the challenge across different student levels, although this would move away from the original intention to specifically target particularly able or previously experienced programmers studying at introductory level.

Finally the tight travel budget removed any opportunities for the teams to socialize informally before of after the event. It was felt that it a useful additional objective would be to encourage members of the different teams to mix and socialize during the period in which they aren't competing – perhaps by staging a fun event of some kind that encouraged interaction.

CONCLUSIONS

The project was successful in its objective of stretching the most able students. However the structure of the competition did not enable all such students at each university to participate in the TOPS activities. The competition introduced higher level activities which required the learners to integrate skills, knowledge and understanding acquired thus far. Usually an introductory programming course would not include these integrated activities in it curriculum.

If we are to offer opportunities created by the TOPS project to all our most able novices, we need to place these opportunities within the standard curriculum, albeit as extension activities. Local heats of the competition prior to the challenge day would make it possible for any of the most able to participate. Perhaps working in an online virtual environment could be used to preserve the inter-institutional structure of the activity for all participants.

As a proof of concept the activities supported by the project appear to be possible. Academics gained valuable insight into curriculum and teaching approaches at other institutions, but the process was not without stress. The inter-university aspect of the challenge was perhaps more important to the academics than the students. Peer observations could be undertaken without the competition, but the academics also learnt from observing their students competing with those from other universities. Cost free video conference meetings might have been used alongside face to face meetings to the benefit of the academics. A small amount of additional funding would have been able to realize its original objectives. The planning team now has a realistic estimate of the amount of cash and sponsorship they should seek.

Members of the project team and the students have identified ways in which the activity might be fine tuned in the future. The value of the activity was recognized by representatives of both the original funding body and the sponsoring organization. The project team is keen to find ways of raising additional funding and are particularly interested in extending the activity to involve a greater number of universities. Such a change will inevitably generate additional administrative load – and if the current round of activity is to be taken as a guide, there would be a level of stress associated with that change. However colleagues and student participants found the first round of the TOPS programming challenge a rewarding and worthwhile activity.

If larger numbers of students were to participate across a greater range of institutions there would be an opportunity to engage in some data gathering which might result in useful additional information and analysis which could helpfully contribute further to the debate which surrounds the perennial discussion of how best to meet the diverse needs of students on introductory programming courses.

ACKNOWLEDGMENT

Work undertaken in this paper was supported by grant funding from the UK Higher Education Academy Subject Centre for Information and Computer Science. The activity was further supported by hosting and prizes from Sun Microsystems.

REFERENCES

[1] Davis, H. C., Carr, L. A., Cooke, E. C., and White, S. A., "Managing Diversity: Experiences Teaching Programming Principles," presented at the 2nd LTSN-ICS Annual Conference, London, 2001.

[2] Carr, L. A., Davis, H. C. and White, S. A., "AnnAnn - a Tool to Scaffold Learning About Programs," presented at the Fifth Annual Conference of LTSN-ICS, Ulster, UK., 2004.

[3] Kölling, M. and Barnes, D. J., "Enhancing Apprentice-Based Learning of Java.," presented at thirty-fifth SIGCSE technical symposium on computer science education,, 2004.

[4] Fincher, S., Barnes, D. J., Bibby, P., Bown, J., Bush, V., Campbell, P., Quintin Cutts, Jamieson, S., Jenkins, T., Michael Jones, Dimitar Kazakov, Lancaster, T., Ratcliffe, M., Seisenberger, M., Shinner-Kennedy, D., Wagstaff, C., White, L., and Whyley, C., "Some Good Ideas from the Disciplinary Commons," presented at the 7th Annual Conference of the Subject Centre for Information and Computer Science, Dublin, 2006.

[5] Carter, J., English, J., Ala-Mutka, K., Dick, M., Fone, W., Fuller, U., and Sheard, J., "How Shall We Assess This?. , . ," *ACM SIGCSE Bulletin*, vol. 35, pp. 107-123, 2003.

[6] Carter, J. and Boyle, R., "Teaching Delivery Issues: Lessons from Computer Science," *Journal of Information Technology Education*, vol. 1, pp. 77-89, 2002.

[7] Cegielski, C. and Hall, D., "What Makes a Good Programmer?," *Commun. ACM*, vol. 49, pp. 73-75, 2006.