Experiences of Prospective High School Teachers Using a Programming Teaching Tool

Louis Major, Theocharis Kyriacou and Pearl Brereton
School of Computing and Mathematics
Keele University
Staffordshire, UK
l.major@epsam.keele.ac.uk, t.kyriacou@cs.keele.ac.uk, o.p.brereton@cs.keele.ac.uk

ABSTRACT
During their time at school few high school students are exposed to basic computing concepts as Computer Science (CS) is not considered to be an important part of the curriculum. As a result many high school students do not encounter programming during their studies. In the UK, despite newly qualified CS and ICT (Information and Communication Technology) teachers having specific subject knowledge, in recent years schools have steadily ‘watered down’ the content of CS courses and have aligned the subject with disciplines such as Business Studies. This has distanced Computing from the other sciences whilst adding to students’ confusion about what the subject actually entails. This paper presents the results of a study which involved 23 trainee CS/ICT teachers. The study took the form of a day-long workshop session and had the objectives of determining what perceptions and opinions the trainee teachers held in regards to programming as well as to discover the effectiveness and potential of a programming teaching tool based on the concept of a robot simulator. Analysis of data collected during the session shows how, despite the majority of trainees having some programming experience, a number of trainees had difficulty completing basic programming challenges. This displays how some CS/ICT teachers may lack fundamental programming knowledge. Moreover, whilst most of the trainees felt programming should be taught to high school students studying CS or ICT course, fewer than half of the trainees said they had the confidence to teach the subject. An evaluation of the effectiveness of the robot simulator as a teaching tool is also presented as is a discussion on the implications which these findings may have.

Keywords
Teaching, programming, simulator, teachers, high school, java

1. INTRODUCTION
Computing differs from many of the traditional sciences as it is not considered to be a core curriculum subject at high school [1]. This has resulted in most high school students not being introduced to Computer Science (CS) during their studies [2]. As a consequence few high school students, even those enrolled on CS and CS-related courses, complete school having encountered important computing concepts. Govender and Grayson in [3] state that effective teachers of programming need detailed knowledge not only about their subject matter but also about the subject matter for teaching. Traditionally, however, few high school teachers who have been in a position to deliver lessons on programming can be considered to be ‘specialists’ in the subject.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Koli Calling ’11, November 17–20, 2011, Koli, Finland.
Copyright 2011 ACM 978-1-4503-1052-9/11/11…$10.00.

For example, in UK schools CS and CS-related disciplines such as ICT (Information and Communication Technology) have often been placed in the same department as subjects such as Business Studies [4]. As a result, many schools have allocated CS teaching duties to staff whose main expertise lies in the teaching of theoretical business concepts rather than practical CS elements. This has resulted in only a few students being introduced to programming. Aligning CS with such disciplines has also further distanced Computing from the other sciences whilst adding to students’ confusion about what the subject actually entails [5].

Today, in order to achieve qualified teacher status in the UK, a prospective high school teacher normally completes a Post Graduate Certificate of Education (PGCE) course. As the main focus of a PGCE is upon developing the ability to teach, not on teaching the subject matter itself, applicants normally possess a relevant qualification (e.g. a degree, conversion qualification or sufficient experience) in the subject they wish to deliver [6]. Such individuals should have practical knowledge (e.g. programming expertise) which enables them to better deliver CS and ICT courses than teachers who have previously specialised in other subjects. In recent years, however, it has been found that UK schools have steadily reduced the CS content of computing courses. This has been replaced instead with ICT training material [7]. As a result, the main focus of CS courses has shifted away from the teaching of practical computing concepts. It appears, therefore, that the ‘watering down’ of CS in high schools has continued despite newly qualified teachers having some level of specific subject knowledge.

Learning to program is challenging for novice programmers and as a result programming courses often have high drop-out rates [8]. Teaching high school students the basics of programming is also difficult due to a lack of expertise, available resources and time. To overcome these issues educators have implemented a variety of techniques which aim to help novice programmers. Such attempts can be referred to as interventions. Many of these interventions employ the idea of active-learning environments and evidence suggests that such a technique promotes better understanding and retention of programming concepts [9].

This paper describes the use of an active-learning environment which can be used to teach introductory programming concepts to novices. This work builds upon the results of a Systematic Literature Review (SLR) which evaluated the effectiveness of using robots as interventions to teach programming [10]. In the SLR it was found that there was the potential to further investigate, and evaluate, the use of simulated robots as programming teaching tools. This is because, out of the 34 papers which were analysed during the SLR, only seven papers were found to examine the use of simulated robots in an introductory programming setting. Such a quantity of papers was not found to
represent a complete body of research and it was concluded that there was scope for further empirical research to be carried out to investigate such an intervention. This paper expands upon the findings of the SLR by using a robot simulator as a tool to teach programming.

In the conclusions to her doctoral thesis Sarah Anne Drummond (see [1]) states that it would be useful to study school teachers’ perceptions of Computing and that it would be interesting to investigate how they regard the discipline. Other work has also reported how trainee teachers usually adopt teaching philosophies and methods based on their own experiences [11]. In this paper a study is reported which involved 23 trainee teachers (hereafter referred to as trainees) enrolled on a PGCE course. The trainees will be qualified to deliver Computing and ICT courses in UK high schools six months after the study took place. The study had two interlinked objectives based upon the conclusions reported by this past literature [1, 11] in addition to addressing the findings of the SLR:

1. To determine what perceptions and opinions the trainees held in regard to aspects of Computing (namely computer programming).

2. To demonstrate and determine the effectiveness and potential of an innovative programming teaching tool, which uses the concept of a robot simulator that can help educate novice computer programmers.

The remainder of this paper is divided as follows. In Section 2 the methodology of the study is presented. This is followed by a results section and a discussion in Sections 3 and 4 respectively. Finally, in Section 5, a conclusion is provided.

2. METHODOLOGY

2.1 Introduction to the Robot Simulator

The robot simulator that was used during the study is modelled after on the Mark III robot which is designed and sold by the Portland Area Robotics Society (PARTS). A Mark III robot is pictured in Figure 1. The simulator has been developed by one of the authors of this paper (TK).

![Figure 1. A photograph of a Mark III robot after which the robot simulator used in this study is modelled (Width 10cm, Length 10cm, Height 8cm).](image1)

The robot simulator is written in Java and a range of introductory programming concepts can be taught using the software. When the simulator is first loaded users are presented with a simple interface and a blank arena into which their robotic agent can be placed. It is possible to introduce 2D and 3D objects into this arena. The task of programming the robots takes place away from the Graphical User Interface in a separate window. Each robotic agent has two actuators in the form of wheels which enables movement around the environment. The creation of interactive programming tasks such as wall-following and maze-solving is possible as the agents used are equipped with several sensors. As a result the robotic agents can be programmed to follow, seek and avoid objects based on user input. In Figure 2 a screenshot of the robot simulator is displayed. For further information on how the robot simulator works, and what the advantages of using such a method are, see [12].

![Figure 2. A screenshot of the robot simulator. Here a robotic agent is navigating around obstacles by using its sensors.](image2)

2.2 Overview of the Study

The study that has been performed involved 23 trainee teachers enrolled on an ICT PGCE course at Keele University. 15 of the trainees were male while eight were female. The trainees will be qualified to deliver ICT and Computing related courses in high schools six months after the study took place. The trainees had already gained practical classroom experience as part of their PGCE. The background of the trainees, in relation to their past programming experience, was varied and this will be discussed in the results section of this paper (Section 3.1.1). The workshop session was hosted in a computer laboratory located in the School of Computing and Mathematics at Keele University. The day-long workshop session took place in April 2011. Upon entering the lab the trainees were allocated to a PC, instructed to read an information sheet and asked to complete a consent form if they were prepared to take part. The information sheet contained details about the nature of the research being performed and the trainees were given an opportunity to ask questions. It was made clear to the trainees that they could withdraw from the study at any time. Once the trainees had agreed to participate in the research a questionnaire was distributed to each trainee. This questionnaire is referred to as the ‘pre-workshop questionnaire’ during this paper. This questionnaire was devised to determine the trainee’s level of past programming experience, to gauge their past enjoyment of programming in addition to ascertaining their thoughts on the current teaching of programming in high schools. All of the questionnaires used during the study were completed anonymously by the trainees and were posted into a sealed box. Following this a presentation of around 45 minutes took place. This introduced the trainees to the robot simulator. The trainees were then issued instructions on how to use the robot simulator...
and its features before being introduced to the BlueJ Integrated Development Environment\(^1\) (which the simulator uses). Once these preparatory steps were completed it was then possible to introduce programming concepts using the robot simulator as a teaching tool.

Figure 3. A robotic agent completing a ‘figure of 8’ – one of several programming tasks the trainees participated in.

A variety of introductory programming concepts were demonstrated using the robot simulator during the workshop. These included (but were not limited to) an introduction to: statements, variables, expressions, sequence, selection, iteration and methods. Using these programming concepts the trainees were asked to solve a range of challenges. These included making their simulated robotic agents complete a ‘figure of 8’, follow objects, avoid objects and seek-out objects (See Figure 3 for an example of a robotic agent completing a ‘figure of 8’). The workshop culminated in the trainees attempting to make their robot follow lines and walls which they had introduced into the robots environment. Two authors of this paper (LM and TK) were on hand to offer assistance to the trainees when they ran into difficulties. Several discussions also took place during the course of the activities and the group was convened several times for discussion. Once all the programming challenges were completed, and the workshop session was over, the trainees were asked to fill in a second questionnaire. This will be referred to as the ‘post-workshop questionnaire’. This questionnaire shared several similarities with the pre-workshop questionnaire. There were, however, several differences and the questions were devised to determine the trainee’s enjoyment and perceived difficulty of their programming experience during the workshop in addition to their thoughts on the effectiveness of the robot simulator as a tool to introduce programming concepts. The session was then ended and the trainees thanked.

3. RESULTS

3.1 Pre-Workshop Questionnaire Results

3.1.1 Trainees Past Programming Experience

In order to gauge their level of programming experience the trainees were asked to select from several options. If a trainee selected ‘I have no past programming experience’ they were instructed to skip the questions related to programming experience and to move on to the second section of the questionnaire (discussed in Section 3.1.2). Participants were also able to specify (in their own words) details about their past programming experience. Only one of the trainees stated that they had no past programming experience. The other 22 participants completed the first section of the questionnaire in full. 19 of the trainees selected ‘I have programmed because a University (or other) course required me to’. Five trainees selected the option that they had previously programmed out of personal interest, three indicated that they had gained programming experience as part of a previous job whilst three said they had previously programmed to learn a new skill. The trainees were then asked to specify what programming languages they were familiar with. In total the trainees responded that collectively they had experience using 22 programming languages. These ranged from languages such as Java and Visual Basic, web-development languages like ActionScript and database query languages such as SQL. The two most popular languages which the trainees had experience using was Java (15 trainees) and VB.Net or VB (9 trainees).

![Table 1. Programming concepts arranged by the number of trainees who have used such a concept in their past programming code](image)

In order to determine the trainee’s familiarity with introductory programming concepts they were then asked to select from a list of concepts which they had previously used in programming code. The results of this can be seen in Table 1. The 22 trainees who had previously programmed were then instructed to specify which of the following statements best describes their past programming experience: ‘Didn’t like programming’, ‘Indifferent’ or ‘Enjoyed programming’. It was found that nine of the trainees stated they had enjoyed their past programming experience while five of the trainees did not. Eight of the trainees said they held an indifferent view towards programming based upon their previous experience. To conclude this section of the questionnaire the trainees were then asked whether they had found their past programming experience challenging. 19 of the trainees selected the ‘Challenging’ option while three selected ‘Neither challenging nor trivial’. None of the trainees responded that they had found their previous programming experience to be ‘Trivial’.

3.1.2 Trainees Opinions on Teaching High School Students Computer Programming

The second section of the pre-workshop questionnaire focused on the trainee’s opinions of teaching programming to high school students. All 23 participants completed this section in full. Firstly the trainees were asked, “Do you believe that it is important to teach basic programming concepts to all high school students enrolled on an ICT or Computing course at some stage during their time at school?” In response to this question 19 trainees selected ‘Yes’ while four trainees chose ‘Not Sure’. A second question was then put to the trainees in order to determine the confidence they had in their current ability to teach programming. The trainees were asked, “With your current knowledge would you be confident teaching high school students about introductory

\(^1\) http://www.bluej.org/ (Accessed 27/09/2011)
programming concepts?”. In response to this question 10 trainees said ‘Yes’, 8 ‘Not Sure’ and 5 ‘No’. Finally the trainees were asked how difficult they felt it would be to teach elementary programming concepts to high school students using their current knowledge. The responses to this question can be seen in Table 2.

Table 2. Perceived difficulty amongst the trainees in teaching elementary programming concepts to high school students using their pre-workshop knowledge

<table>
<thead>
<tr>
<th>Questionnaire Option</th>
<th>No. of Trainees</th>
</tr>
</thead>
<tbody>
<tr>
<td>Difficult</td>
<td>14</td>
</tr>
<tr>
<td>Neither Difficult nor Easy</td>
<td>6</td>
</tr>
<tr>
<td>Not Sure</td>
<td>2</td>
</tr>
<tr>
<td>Easy</td>
<td>1</td>
</tr>
</tbody>
</table>

3.2 Post-Workshop Questionnaire Results

3.2.1 Trainees Opinions of the Robot Simulator as a Tool to Teach Programming

To gauge the trainees enjoyment of their programming experience using the robot simulator they were asked, “In regards to today’s programming experience, which of the following best describes your enjoyment of today’s session?”. 22 of the trainees stated that they had ‘Enjoyed’ the session whilst only one selected ‘Not Enjoyable’. The trainees were then questioned on how challenging they felt the programming tasks set during the workshop had been. Three responses were available to this question: ‘Easy’, ‘Neither easy nor difficult’ and ‘Difficult’. None of the trainees selected the easy option while 11 of the trainees stated that they found the programming tasks neither easy nor difficult. The remaining 12 trainees chose the difficult option. To assess the trainees thoughts on the robot simulator the following question was used, “Do you believe that the robot simulator offers an effective method of introducing basic programming concepts, which you have been taught today, to novice programming students?”. 21 of the trainees responded ‘Yes’ while two trainees responded ‘Not Sure’. Following this the trainees were then asked whether they would consider using the robot simulator as a tool to teach programming in their own future lessons. 22 participants responded that they would while one said they were unsure. In order to establish how effective different elements of the workshop had been, in relation to the teaching of programming, the trainees were then instructed to rate on a scale of one (not at all effective) to five (extremely effective) the effectiveness of: the simulator, teaching delivered during the session, the presentation delivered during the session and the teaching environment. The responses supplied by the trainees can be seen in Table 3.

Table 3. Trainee opinions of how effective different elements of the workshop were (1 = not at all effective, 5 = extremely effective). Arranged by number of trainees.

<table>
<thead>
<tr>
<th>Workshop Element</th>
<th>1 (Not at all effective)</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5 (Extremely effective)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulator</td>
<td></td>
<td>10</td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Teaching</td>
<td></td>
<td>5</td>
<td>5</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>Presentation</td>
<td></td>
<td>1</td>
<td>4</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Environment</td>
<td></td>
<td>3</td>
<td>2</td>
<td>11</td>
<td>7</td>
</tr>
</tbody>
</table>

3.2.2 Trainees Post-Workshop Opinions on Teaching High School Students Computer Programming

The post-workshop questionnaire also contained two questions which were used in the pre-workshop questionnaire. These two questions were re-used in the post-workshop questionnaire to see if the workshop session had any effect on the trainee’s confidence in their ability to teach programming. As before the trainees were asked, “With your current knowledge would you be confident teaching high school students about introductory programming concepts?”. 13 of the trainees reported that they would be confident doing this and selected ‘Yes’. Four of the trainees suggested they would not have the confidence and they chose ‘No’. Six of the trainees selected ‘Not Sure’. This was followed by the trainees being questioned on how difficult they felt it would be to teach elementary programming concepts to high school students using their post-workshop knowledge. In response 13 of the trainees said that it would still be ‘Difficult’, seven that it would be ‘Neither difficult nor easy’ and two selected the ‘Easy’ option. One of the respondents indicated that they were ‘Not Sure’.

3.2.3 Thematic Analysis of the Trainee’s Opinions of the Robot Simulator

The final three questions of the post-workshop questionnaire aimed to identify which aspects of the robot simulator the trainees liked the most, liked the least in addition to the gathering the trainees general opinion of the workshop session. These questions were ‘open’ questions whereby the participants could respond in their own words. In order to analyse the data collected from the responses a thematic analysis has been undertaken. This was done by identifying common trends in the trainee’s replies and by categorising this information into groups. The thematic analysis of the data has resulted in several trends being identified. These are illustrated here by using examples of quotes which the participants provided. In order for a trend to be included in the thematic analysis, at least two trainees must have made reference to the trend in question.

The idea that the robot simulator was enjoyable to use was mentioned by several of the trainees with comments such as, “Fun and a good challenge”. The visual and tangible nature of the robotic agents also appeared to help the trainees engage with the simulator and programming tasks. “The simulator provides visual feedback on erroneous code which is a very useful tool to identify and correct errors” and “… you can see the end results to see if changes need to be made”. How the simulated agents, “… allowed you to see easily what your programming did” was also mentioned. The fact that high school students could get immediate results from the simulator and that the programming ideas which underpin the software would, “… challenge the pupils” was also commented on. Four of the participants also remarked how the simulator and its interface helped to make programming concepts more understandable and that it was good actually seeing programming and its effects. Two main concerns were highlighted by the trainees when asked which aspects of the robot simulator they liked the least. Firstly, several ‘flaws’ of the simulator were identified including the absence of an undo feature, that some of the variable names used by the simulator are slightly abstract and that a help or frequently asked questions feature may be beneficial. Secondly, several of the trainees highlighted their concerns about the suitability of the simulator for all high school students no matter what their ability. One trainee responded that Java, “… was a complex language...
particularly for low literacy students”. Another suggested they were, “... not sure whether the tasks would be too difficult for lower Key Stage 3 students (aged 11 – 14)”. Despite these issues the overall opinion of robot simulator and the workshop session was extremely positive. One trainee commented how the workshop was, “... excellent – best session we have had on the PGCE” while another stated how the simulator ‘links well to teaching A-Level ICT’ (A-Levels are the UK’s main post-high school and pre-University qualification). The workshop itself was also described as being a, “... very good session... I would use many of the ideas to present to pupils in my class”. In regards to the teaching of programming one participant said how they had, “... struggled with Java previously but the (session) was well taught and (the simulator) engaging”. Another respondent also commented that they would have liked, “… to know how to get the program onto robots”. These sentiments were shared by several of the other trainees.

4. DISCUSSION

As stated in Section 1, this study had two interlinked objectives based upon the conclusions reported by past literature in addition to addressing the findings of a previously completed SLR:

1. To determine what perceptions and opinions the trainees held in regard to aspects of Computing (namely computer programming).

2. To demonstrate and determine the effectiveness and potential of an innovative programming teaching tool, which uses the concept of a robot simulator that can help educate novice computer programmers.

In relation to Objective 1 it was found during the study that 22 out of the 23 participants had programmed at some point before the workshop. Of the 22 trainees who had programmed previously, 15 had experience with the Java programming language. As a result over two-thirds of the trainees had used Java, the same language that the robot simulator employs. Using the robot simulator as a teaching tool, only basic programming challenges were set during the workshop. As stated in Section 2.2, these tasks related to statements, variables, expressions, sequence, selection, iteration and methods. Before conducting the study it was expected by the authors that those trainees who had previously programmed would have little difficulty recalling the basic, but fundamental, programming concepts being taught. It was also thought that the ‘introductory lecture’, which the workshop commenced with, would serve to re-familiarise the trainees who had programmed before with the concepts they would be using. The results of the post-workshop questionnaire, however, suggest that these assumptions were incorrect as the majority of trainees found completing the tasks set during the workshop difficult. This is despite the bulk of the trainees stating in the pre-workshop questionnaire that they were familiar with the concepts that were later taught to them. Moreover, it is in spite of the trainees collectively having used 22 different programming languages in the past. It is suggested that trainees found the tasks set during the workshop difficult due to a lack of understanding in regard to the programming concepts that were taught. The reasons for this are unknown but it may be because the trainees had not programmed for a long time or because they never truly mastered the programming concepts in the first place. It may also highlight how those trainees who complete a conversion course, as a condition of enrolling on the PGCE, lack basic programming ability. This assumption has been made as several of the participants commented to the authors that they had gained access to the PGCE via a conversion course (i.e. they did not have a first degree in Computing and needed to re-specialise before being able to enrol on the PGCE). Potentially the programming ability of these trainees may not be as advanced as their colleagues on the course who have a Computing undergraduate degree. Further investigation would have to be undertaken to verify such assumptions, however, as neither questionnaire was designed to address such a topic.

In regards to the teaching of programming in high schools 19 of the trainees thought programming should be taught to Computing and ICT students while four were unsure. This demonstrates how the trainees, on the whole, believe programming is an important subject for Computing and ICT students to learn. The pre-workshop questionnaire found, however, that fewer than half of the trainees would be confident teaching basic programming concepts to high school students. Such a finding is a cause for concern as it displays how some trainee CS/ICT teachers lack programming knowledge and confidence in their ability to teach it. As the main aim of a PGCE is to develop the ability to teach, not to enhance subject knowledge, it is fair to assume that the trainees programming ability would not change significantly before they obtained qualified teacher status.

In relation to Objective 2 the results of the post-workshop questionnaire demonstrate how the trainees almost unanimously enjoyed their programming experience using the robot simulator. Likewise, 21 participants stated that they believe the simulator offers a valuable method of teaching basic programming concepts. In terms of effectiveness all of the trainees awarded the robot simulator a score of either a ‘4’ or ‘5’ (with five being the highest achievable score). Despite the majority of participants believing that the simulator was an effective tool, however, there was only a moderate change in the trainee’s confidence in their ability to teach high school students programming as a result of the workshop session. Similarly, the trainee’s opinions on how difficult they thought it would be to teach elementary programming concepts to high school students did not change significantly over the course of the workshop. It is interesting to note that the trainee’s confidence and perceived difficulty of teaching programming did not significantly change, despite them believing the robot simulator offered an effective and enjoyable method of learning the subject. The length of the workshop session must be considered, however, as this only took place over the course of one day. As a result, only limited coverage of material was possible and it is perhaps unsurprising that significant changes, in regards to the trainee’s pre and post-workshop confidence, were not reported.

Finally, from the qualitative thematic analysis that was undertaken, several common trends were identified by analysing the trainee’s responses. These included the observations that the robot simulator is fun and effective to use, offers strong visual feedback, helps to demonstrate the effects of changing code, makes it easy to identify and correct programming errors and offers a good programming challenge. Several shortcomings of the simulator (and potential improvements) were also highlighted by the trainees in their responses. It is intended that the feasibility of addressing these issues will be investigated as the simulator evolves. Of greater concern to the authors, however, is that some participants questioned the suitability of the robot simulator for use by high school students of all abilities. It is considered that the reason for such comments is due to the nature of the simulated
robots that are currently used. Potentially the use of simpler agents initially, within a more restrictive environment may help to overcome such concerns.

5. CONCLUSION AND FUTURE WORK
This paper has reported on a study which involved 23 trainee CS/ICT teachers and took place at Keele University, UK. The study took the form of a day-long workshop session and had the objectives of determining what perceptions and opinions the trainees held in regards to computer programming as well as to discover the effectiveness and potential of a programming teaching tool based on the concept of a robot simulator. Two questionnaires were completed during the study (one before the workshop and one after). This study helps to address in part the findings of previously completed research, namely a Systematic Literature Review, which suggested how further research was required in order to investigate the effectiveness of simulated robots as programming teaching tools.

It was found during the study that the majority of trainees had programmed before and collectively the trainees had experience using 22 different programming languages. In addition 14 of the trainees had previously used the Java language, the same one that the robot simulator employs, at some point before the workshop. Despite these factors, however, a number of trainees had difficulty completing the basic programming challenges set during the workshop. This displays how some trainee CS/ICT teachers lack the basic knowledge required to teach programming. Moreover, whilst most of the trainees believe CS and ICT high school students should learn programming, fewer than half of the trainees felt they had the confidence to teach the subject. This highlights how, if programming is to become an established part of the high school curriculum then much needs to be done to inform prospective teachers so they feel they are able to teach the subject adequately. Furthermore, the robot simulator that was implemented was found to offer an enjoyable and effective method of teaching programming. This is despite the simulator only moderately improving the trainee’s confidence in their ability to teach programming.

As a final thought, it is the belief of the authors that if programming is ever to become an established part of the high school curriculum then much needs to be done to prepare teachers so that they are confident and able to teach the subject. Whilst the study that has been presented is only modest in terms of scale, several important findings have been discussed. Workshops that serve to inform high school teachers on the best practice to teach programming, in a manner similar to the one described, are a positive step in the direction of making programming a recognised high school subject. Such workshops also offer an opportunity to inspire teachers to teach subjects which they may not ordinarily consider delivering. Ultimately, however, it will likely be the policy makers (such as Local Education Authorities and Central Governments) who will dictate whether programming will become an established part of a countries high school curriculum. At present teachers and school managers are content to teach subjects that they know well. Regular school inspections and an increase in bureaucracy serve to discourage innovative teaching practices and the introduction of new material by the teachers themselves. This may be because the teacher’s fear they will be penalised (e.g. by being awarded a low teaching score during school inspections) if they take a risk and teach a topic such as programming as they only have partial understanding of the subject matter. If programming is ever to become a recognised curriculum subject, therefore, the support of policy makers will likely first have to be achieved in addition to the task of better informing the teachers themselves.

6. ACKNOWLEDGMENTS
The authors would like to thank Bruce Nightingale of Keele University for his assistance in helping to set up the workshop session in addition to the trainees who participated in the study.

7. REFERENCES