# Chapter 7
# Problems First

**Gary Hill**
*University of Northampton, UK*

**Scott Turner**
*University of Northampton, UK*

## ABSTRACT

*This chapter considers the need to focus initial programming education on problem-solving, in advance of programming syntax and software design methodology. The main vehicle for this approach is simple Lego based robots programmed in Java, followed by the programming of a graphical representation/simulation to develop programming skills. Problem solving is not trivial (Beaumont & Fox, 2003) and is an important skill, central to computing and engineering.*

*An approach will be considered, illustrated with a series of problem-solving tasks that increase in complexity at each stage and give the students practice in attempting problem-solving approaches, as well as assisting them to learn from their mistakes. Some of the problems include ambiguities or are purposely ill-defined, to enable the student to resolve these as part of the process.*

*The benefits to students will be discussed including students' statements that this approach, using robots, provides a method to visually and physically see the outcome of a problem. In addition, students report that the method improves their satisfaction with the course.*

*The importance of linking the problem-solving robot activity and the programming assignment, whilst maintaining the visual nature of the problem, will be discussed, together with the comparison of this work with similar work reported by other authors relating to teaching programming using robots (Williams, 2003).*

DOI: 10.4018/978-1-60960-797-5.ch007

## INTRODUCTION

This chapter considers the teaching of programming and problem solving to undergraduate first year computing students, using robots and graphical programming to emulate the robot tasks.

Probably the most important skills a computer scientist or engineer must possess are those of problem solving. These skills are highlighted in numerous benchmark and guideline statements for engineering and computing (Adams et al., 2008). While it is appreciated that being a good problem solver involves knowledge and experience, there are other interventions, such as training and practice, that can improve process skills in engineering and computing undergraduates.

Mindstorm based robots have been used previously for teaching programming to computing and engineering students (Fagin, 2003; Lawhead et al., 2003; Price et al., 2003; Williams, 2003). Here we make use of them for problem-solving. Synergies can be achieved using robots to develop problem-solving skills and skills of pre-object programming (for example Culwin et al., 2006) and simulation of robots for teaching programming as a visual approach in the teaching of the widely used programming language Java.

The approach discussed here focuses upon the development of problem-solving skills first and not on learning a new programming language from the outset. Therefore, initially, any programming is kept simple with the minimum of commands, with *objects* unknowingly used, as these are later introduced/learned during the programming stage of the computing module. Work within the team (Turner & Hill, 2008) suggests that using Lego robots within the teaching of problem-solving and the resulting java GUI emulation has some benefits for the students. The students come on the courses with a range of experiences and abilities, but many of them have limited or no experience of programming. In other words, this is likely to be the first time they are exposed to programming or expected to program. It was felt within the course team that giving the students a tool that is non-threatening and that would also provide early success in a physical and visual form (which, if it were to break, it would not matter), would be beneficial. Lego-based robots are initially used, followed by the same problems being repeated as a graphical representation/simulation to satisfy these same criteria.

Other authors have noted the suitability of robots for teaching programming. Lawhead et al. (2003) stated that robots provide entry level programming students with a physical model to visually demonstrate concepts, that the most important benefits of using robots in teaching introductory courses is the focus provided on learning language-independent truths about programming and programming techniques and that robots readily illustrate the idea of computation as interaction. This chapter advances the idea that developing problem-solving skills first, then embarking on programming, is beneficial. We explore this idea in detail by discussing the module where these ideas have been tested.

The module is structured into two parts, eight weeks (16 hours) spent on problem-solving, followed by sixteen weeks (32 hours) of graphical programming in Java. The underlying approach is that as the module develops the focus evolves from general concepts of problem-solving (e.g., brain-storming, functional decomposition) to solving problems based around robots which increase in difficulty (but not necessarily in complexity). The module is first assessed by a robot-based project, which then leads into developing the same problem via a graphical user interface in Java, which is finally assessed. The authors believe that the visual nature of the work and the linkage of the assignments aid the development of the necessary skills.