

# Establishing Structured Support for Programming Students

Ioanna Stamouli<sup>1</sup>, Eileen Doyle<sup>2</sup>, and Meriel Huggard<sup>3</sup>

**Abstract** – This paper describes the structure, implementation, and impact evaluation of a Programming Support Centre for engineering and computing students. The main focus of this centre is to provide a positive, supportive atmosphere where students can voluntarily seek one-to-one assistance with programming difficulties. The support offered is specifically structured to nurture and leverage each student's motivation for taking a programming course whilst providing them with individually tailored advice and practical help. A qualitative and quantitative evaluation of the centre's operation is presented, together with analysis of statistics on student motivation. The results of this research suggest that the newly developed programming support centre has had a positive impact on student learning.

*Index Terms:* Programming support, introductory computer programming, student motivation.

## INTRODUCTION

Computer programming is a fundamental building block of a computer science curriculum. Many students encounter difficulties in their first year programming courses and, as graduates, display little or no confidence in their programming skills. The nature of the computer science curriculum is such that there is very little continuity between subjects studied during the traditional secondary education senior cycle and those encountered during the first year of undergraduate studies. Indeed, four of the courses offered as part of a first year degree programme in computer science in Trinity College, Dublin: Communications, Computer Science (Programming), Computer Technology and Information Management; are beyond the scope of the students' previous experience. These courses require a style of logical thinking that many students have not experienced prior to entry to university. Many of these students are experiencing university life for the first time and haven't developed an appropriate sense of their own responsibility for, or ownership of, their learning.

There is little doubt that students frequently experience problems with introductory programming courses: many engineering and computer science students claim to 'hate programming' and feel unable to grasp even the most basic

skills [15]. Students who struggle with programming courses either fail and drop out of the course; or manage to continue but assiduously avoid future programming projects and ultimately choose a career path that does not involve programming.

Many researchers recognise that learning to programme depends on motivation, aptitude, and multiple other skills, such as critical thinking and mathematical ability. To address an individual student's problems with programming requires structured support that motivates their learning and helps them to adopt appropriate learning styles.

A recent initiative established a Programming Support Centre (PSC) within the Department of Computer Science, Trinity College. The main feature of the centre is the provision of one-to-one support and resourcing to assist students with any programming problem that they may be facing. The centre staff seek to help and encourage the students to become motivated, independent learners, and therefore to increase their possibilities for academic success.

In the following sections factors that affect the success of novice programmers e.g. aptitude and motivation, are discussed. The direct influence of such factors on the establishment of a Programming Support Centre to support student learning is then described. Finally, a qualitative and quantitative evaluation of the centre is presented: this includes usage statistics and an assessment of students' perceptions of their programming difficulties. In-depth analysis results in recommendations on ways that students programming endeavours can be supported in a structured environment.

## SUPPORTING NOVICE PROGRAMMERS

To provide structured support for programming students it is necessary to establish the factors that directly affect their success on programming courses. Two key factors which have been identified are student aptitude [7] and motivation [13].

### *Aptitude*

The difficulty experienced in learning to program varies for each individual student. Some studies hypothesize that students who find programming difficult are those who have no aptitude for programming [7]. A wide variety of metrics, which seek to determine the aptitude of a student for programming exist, but it has proven difficult to provide evidence for their effectiveness [10]. In general, these are

<sup>1</sup> Ioanna Stamouli, Department of Computer Science, Trinity College Dublin, [stamouli@cs.tcd.ie](mailto:stamouli@cs.tcd.ie)

<sup>2</sup> Eileen Doyle, Department of Computer Science, Trinity College Dublin, [eidoyle@cs.tcd.ie](mailto:eidoyle@cs.tcd.ie)

<sup>3</sup> Meriel Huggard, Department of Computer Science, Trinity College Dublin, [Meriel.Huggard@cs.tcd.ie](mailto:Meriel.Huggard@cs.tcd.ie)

administered prior to course commencement and are usually of a mathematical nature [1]. Recent work in this field has proved inconclusive. In [3] it is suggested that a relationship exists between programming skills and experience in mathematics, while a study at the University of Leeds [4] which streamed students based on the results of an aptitude test, concluded that no relationship existed between the final result in programming and the measured aptitude. Other work has shown that demographic factors are not suitable predictors of success in programming [5]. In summary, it may not be possible to directly determine student aptitude for programming and hence use this to guide the provision of support services for novice programmers.

### Motivation

Student motivation is a key issue in the learning process; however determining what motivates an individual is not trivial. A study of introductory science courses concluded that many students give up and turn away from science due to the failure of introductory courses to motivate students. Participants claimed that the course environment forced them into passivity and competitiveness [16].

Biggs categorises motivation into four classes [2]:

- Extrinsic motivation: focuses on the outcome of the effort (i.e. a good future career).
- Social motivation: relates to what other people value (i.e. friends and family).
- Achievement motivation: the opportunity provided for ego-enhancement. (e.g. students boost their egos by competing).
- Intrinsic motivation: the process of doing is central (e.g. a genuine interest in the subject).

An individual's primary motivation appears to be a significant influence on course completion [6]. In [13], Sheard points out that student's who have major problems with programming are more likely to be extrinsically motivated while their colleagues that excel usually fall into the other motivational classes. Further, students generally retain some form of motivation throughout their programming course [7].

Motivation is often viewed as a product of two factors (see, for example, [2]): *expectancy* and *value*. A student must both expect to succeed and value their success. Based on these criteria it is noted in [6] that 'if a student does not expect to pass a programming module, then, no matter how they value success, they will not be motivated, and will not engage in tasks that the teacher devises'. Moreover, the criteria students' use to measure success can vary widely: some students may be content to pass their programming courses while others seek to obtain high class results. A student's motivation for taking a programming course must be nurtured and leveraged to make them value a successful outcome (i.e. passing the course).

### Other Skills

Being able to program is not a single skill but it is rather a set of skills that form a hierarchy [7, 14]. Programmers utilise some, or all, of these skills at any given point in time. One such skill that is considered essential is critical thinking. This can be defined as a "disciplined, self-directed thinking which exemplifies the perfection of thinking appropriate to a particular domain or mode of thinking" [11]. Other skills required in coding are knowledge of the programming language's basic syntax, the language semantics and the ability to structure a program. All these are essential skills that a student should possess in order to be able to successfully program.

### STRUCTURED PROGRAMMING SUPPORT

Concern over the decline in students' mastery of basic mathematical skills led several U.K. universities to establish Mathematics Support Centres [12]. These enable students experiencing difficulties with mathematics to obtain supplementary resources and tuition. These centres provide support for students of mathematics, computing and engineering where a high level of mathematical skill is required. One of the key features of such centres is that attendance is voluntary, but students experiencing difficulties are strongly encouraged to make use of the facility. Such centres have proved very successful: in a recent survey 61% of centres rated their success as a grade seven or higher (on a scale of one to ten, ten being extremely successful) [8, 9].

The Trinity College Programming Support Centre (PSC) adapts the methodology behind such support centres for use with students taking programming courses. The service offered complements existing lectures, tutorials, seminars and problem classes. A key feature of the PSC is the provision of structured, one-to-one support tailored to each individual student's needs. The centre is not a service to replace lectures or tutorials: support and advice are given on the basis that students have demonstrably made attempts to tackle their specific problems prior to visiting the centre.

### AIMS OF THE PROGRAMMING SUPPORT CENTRE

The objective of the Programming Support Centre is to motivate students to improve their programming skills and enhance their personal academic success; it is hoped that in the longer term this will impact on course retention and completion rates. The centre's aims are to:

- Ease the transition of students into programming courses.
- Motivate students to adopt a positive attitude towards programming and their studies in general.
- Develop the students' programming and critical thinking skills through advice and practical help.

*Environment*

The centre provides a positive, supportive atmosphere to promote student learning. The centre is open for sixteen hours per week, at times that are carefully chosen so that the majority of students are able to avail of its services; typically around lunch time and after lectures in the afternoon. The opening times are flexible to support students prior to, and during, examination periods.

The centre is a dedicated room, equipped with a number of work stations, printer, scanner, study area, whiteboards, and small library. Reference materials in the form of books and summary sheets are provided for student consultation. The summary sheets, created by the centre staff, highlight key concepts of the main programming languages taught.

The primary resource offered by the PSC is its staff. Each staff member works with individual students to provide tailored advice and continuity of support. If a student is unable to visit the centre they have the option of emailing the staff regarding their query. The staff members are experienced programmers and remain up-to-date with the latest pedagogical developments for teaching programming. They are also responsible for evaluating the centre and its services; collating data on the centre's performance, highlighting its impact on student learning and suggesting how the centre should evolve to best meet the needs of students.

*Publicity*

As the centre is a relatively new venture it has been necessary to widely publicise its existence to students, faculty members and support staff within the university. The following methods have been employed for advertising the centre and its services:

- Student induction days at the beginning of the first term were used to introduce the centre and explain its role.
- Programming lecturers provided details and opening hours at the beginning or end of each class.
- Posters were placed on notice boards, in laboratories and in communal student areas.
- A dedicated web site details the centre's aims, services, location and contact information.
- Class mailing lists are used to regularly provide key PSC details to students.

**EVALUATION**

The effectiveness and impact of the Programming Support Centre on student learning was evaluated both qualitatively and quantitatively: quantitative data was gathered via questionnaire while comment cards and archival information on student visits was used to obtain qualitative data. The primary goals of the evaluation were to:

- Identify the successful services offered by the centre.
- Establish which services require further development.
- Suggest how the facility can be improved.

The evaluation also sought to gather information on student motivation, as this plays a significant role in student success on programming courses.

Over 300 students from the Faculty of Engineering participated in this survey. These were drawn from first and second year undergraduate students on the degree programmes in Computer Science, Engineering, and Management Science and Information Systems Studies.

The questionnaire was structured to capture information on both students' programming experience, perceived programming problems and motivations. The remaining questions related to the student's experience of the Programming Support Centre.

The centre opened at the start of the academic year in October 2003. During the 3 terms of operation the centre has received 363 visits.

**ANALYSIS**

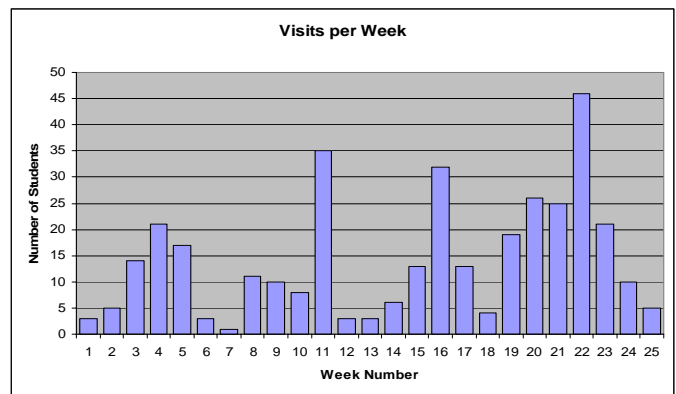


FIGURE 1  
DISTRIBUTION OF STUDENTS VISITS TO THE PSC OVER TIME

From Figure 1, we observe a number of sharp peaks in the distribution of student visits. In the first couple of weeks of operation, most visits to the centre were from interested final year students seeking help prior to commencing their dissertation projects. The first peak in student visits was detected during weeks 4 and 5; this coincided with the first publicity events for the centre. The drop in numbers in week six may be attributed to the December holiday break. It should also be noted that many students had a programming assignment due in week five. The largest concentration in student visits was observed in week 10, again due to programming course assignment deadlines.

In addition to the database statistics presented above, the results from the survey highlights that 26.6% of students targeted have visited the programming support centre. As illustrated in Table I, only 8.6% of the first years questioned had visited the centre. This was a rather unexpected result since it had been anticipated that most of the visits to the programming support centre would be from first year students.

TABLE I  
STUDENT VISITS TO THE PSC

Year	I have visited the PSC	No I haven't
First	8.6%	91.4%
Second	28.5%	71.5%
Total	26.6%	73.4%

However, analysis of the questionnaires given to second year students provided a clue to the lower than expected number of first year visits. In response to the question “*When did you realise you had a problem with programming?*”, 25.6% of students experiencing programming problems had identified this during the first semester of the first year, with a further 53.6% identifying their problem in the latter stages of first year. This could partly explain the low levels of participation by first year students: many have yet to feel they are experiencing major problems with programming.

From Figure 2, we observe that the majority of those who visited the programming support centre were in their second and third year of study. The final category, postgraduate and other, includes one-year students visiting the university on exchange programmes.

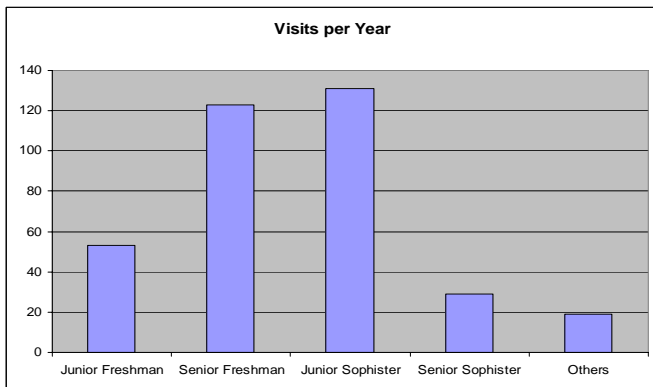


FIGURE 2

DISTRIBUTION OF STUDENTS VISITS TO THE PSC ACROSS YEARS

Other results suggest that 80% of students prefer to ask their lecturer or a friend when they encounter programming problems. Thus one challenge for the Programming Support Centre is to ensure that students consider it as a source of positive assistance and an accessible resource when experiencing programming problems.

Table II presents a summary of the reasons provided by students for not using the centre. The most interesting finding was that in spite of our major publicity efforts 44% claimed that they did not know about the facility.

TABLE II  
REASONS FOR NOT VISITING THE PSC

Reasons	Percentage
I didn't know about the PSC	44%
I didn't require any programming help	26%
I didn't know where it is located	11%
The timetable does not suit me	8%
Other	11%

This survey also investigated the possible motivations that students have towards programming. The students were asked to select their three key motivations from a list that followed Biggs classification. The majority of students are extrinsically motivated: 18% of those surveyed replied that their driving force is a successful future career. However, there is a significant difference between first and second year students: only 13.1% of second year respondents were extrinsically motivated, in comparison to 20.8% of first year students. This may be attributed to the fact that second year students have developed more realistic, mature expectations. The next two most significant motivations were intrinsic as they were related to a genuine interest in learning. These were the desire to learn to program and to improve general problem solving skills. This was a positive outcome, as intrinsically motivated students are more likely to be successful.

As a further evaluation instrument, anonymous comment card were utilised. All visitors to the programming support centre were invited to fill in these cards to provide information on their opinion of the services offered by the centre. The two open questions that are most relevant to this study were “*How helpful was the service provided by the PSC?*” and “*What do you think should be changed or added to the service?*” The comments received were generally positive and helpful to the future direction of the centre. Some example of students comments are below:

- “They explained the problem to me in a logical and straightforward manner”
- “This is a valuable service which should be maintained for the benefit of students”
- “It was very educational”.

**FUTURE DEVELOPMENTS**

The evaluation identified three areas that require further development: publicity, extra staff (to provide extended opening hours) and web based resources. At present (see table II) 44% of the students claim that they were unaware of the existence of the programming support centre. This is unexpectedly high given the considerable time and effort already invested in advertising the centre and its facilities. A more concerted publicity campaign will be undertaken to increase student awareness in regard to the centre. Further comments about the centre suggested that longer opening hours would be appreciated. At present, the centre's opening times are based on the course timetables. However, in order to adopt this suggestion additional funding would be needed so that more staff could be employed. Finally, and most significantly, students recommend the development of tailored online programming resources which they could use when the programming support centre is unavailable.

**CONCLUSION**

A Programming Support Centre has been established in the Department of Computer Science, Trinity College, Dublin. This provides an approachable, non-judgemental service to support students experiencing programming difficulties. Early

findings have been positive and the initiative has been well received by the students.

The results of the evaluation suggest that we have attained our key aims and are progressing towards achieving our long term goals. The number of visits and comments from the students are encouraging. Many of the students who were unaware of the centre's existence also stated that they would consider using its facilities in the future.

There is still scope for improvement. It is apparent that more concerted efforts are needed to publicise the centre. Some of the improvements suggested by students e.g. the web based services, are in hand and are being evaluated at present.

### ACKNOWLEDGMENT

This work has been assisted by the Information Technology Investment Fund, administered by the Higher Education Authority, Ireland. Ioanna Stamouli is supported by the Irish Research Council for Science, Engineering and Technology: funded by the Irish National Development Plan.

### REFERENCES

- [1] Bateman, C. R. "Predicting Performance in Basic Computer Course", in *Proceedings of the 5<sup>th</sup> Annual Meeting of the American Institute for Decision Science*, 1973.
- [2] Biggs J. "Teaching for Quality Learning at University", Society for Research into Higher Education & Open University Press, 1999.
- [3] Byrne P., and Lyons G. "The effect of Students Attributes on Success in Programming", in *Proceedings of the 6<sup>th</sup> Annual Conference on Innovation and Technology in Computer Science Education*, 2001, pp. 49-52.
- [4] Davy J. and Jenkins T. "Research-led Innovation in Teaching and Learning Programming", in *Proceedings of the 4<sup>th</sup> Annual SIGCSE/SIGCUE ITiCSE Conference on Innovation and Technology in Computer Science Education*, 1999, pp. 5-8.
- [5] Evans G., and Simkin, M. "What best Predicts Computer Proficiency?", *Comm. ACM*, Vol. 32, Issue 11, 1989, pp. 1322-1327.
- [6] Jenkins T. "The Motivation of Students of Programming", *Proceedings of the 6<sup>th</sup> Annual Conference on Innovation and Technology in Computer Science Education*, 2001, pp. 53-56.
- [7] Jenkins T. "On the difficulty of learning to program", in *Proceedings 3<sup>rd</sup> LTSN for information and Computer Science Conference, Learning and Teaching Support Network for Information and Computer Science*, 2002, 65-71.
- [8] Lawson, D, Croft, A. C. and Halpin, M. "After the diagnostic test – what next? Evaluating and enhancing the effectiveness of mathematics support centres – part 1", *MSOF Connections*, Vol. 1, No. 3, 2001, pp 19-23.
- [9] Lawson, D, Croft, A. C. and Halpin, M. "After the diagnostic test – what next? Evaluating and enhancing the effectiveness of mathematics support centres – part 2", *MSOF Connections*, Vol. 2, No. 1, 2002, pp 23-26.
- [10] Mazlack L. J., "Identifying Potential to Acquire Programming Skill", *Comm. ACM*, Vol. 23, Issue 1, 1980, pp. 14-17.
- [11] Paul R. W. "Critical Thinking in North America: A New Theory of Knowledge, Learning and Literacy", *Argumentation* 3, 1989, p. 214.
- [12] Savage, M and Hawkes, T. "Measuring the Mathematics Problem", The Engineering Council, London, U.K., 2000.
- [13] Sheard J. and Hegan D., "Our Failing Students: A study of a Repeat Group", *Proceedings of the 3<sup>th</sup> Annual Conference on Innovation and technology in computer science education*, 1998, pp. 223-227.
- [14] Sloane K. D., and Linn M. C., "Instructional Conditions in Pascal Programming Classes", in Mayer R. E. *Teaching and Learning Computer Programming*", Lawrence Erlbaum Associates, 1988, pp. 207-235.
- [15] Thomas L., Ratcliffe M., Woodbury J. and Jarman E., "Learning Styles and Performance in the Introductory Programming Sequence", In *Proceedings of 33<sup>rd</sup> SIGCSE technical symposium*, 2002, pp 33-37
- [16] Tobias S. "They're Not Dumb, They're Different: Stalking the Second Tier", Science News Books, 1994.