

## **An Investigation of the Role Programming Support Services Have for Mature Students**

**Keith Nolan, Amy Thompson, Mark Noone, Aidan Mooney**

Department of Computer Science, Maynooth University, Ireland.

---

### ***Abstract***

*Programming support services for introductory programmers have seen a rise in popularity in recent years with third level institutions around the world providing “safe spaces” for students to practice their programming skills and get supports without the risk of being judged by anyone. These services appear in many different structures including Support Centres, Software Studios and help desks. The common trend however is that all the users of these services, in general, report that the service has helped them in their studies and garnered them with more confidence in their ability.*

*This paper examines the role which our Computer Science Centre played for students who attended the support service during an intensive higher diploma course. The intensive course is a 3-week course tailored to students who have previously completed a degree in a field not related to CS and covers CS1 and CS2 material. The structure and design of the support service is outlined in this paper along with the supports offered. A high-level survey was conducted to investigate the effect of the service on students programming self-efficacy. Study design and methodology are described in detail. Early findings suggest that the support services offered to these students improved their belief in their own programming ability which in turn improved their exam grade outcome. The findings provide valuable evidence to justify future research into the functions of support services with the computer science domain.*

**Keywords:** *Support services, CS1, Self-efficacy, mature students.*

---

## **1. Motivation**

First year computer science (CS) currently boasts one of the highest attrition rates in Ireland with about 25% dropping out, the highest among all higher education disciplines (Harmon & Erskine, 2017). While many possible causes exist, it is well known that first year computer science students generally have little to no formal exposure to CS prior to their CS1 course, which in turn leads to certain concepts in the CS1 course being difficult and hard to grasp (Connolly & Murphy, 2005). In addition to there being difficult concepts, the workload required by students to learn how to program competently can be significant. This workload coupled with the high workload of university, and the fact that CS is often taken as part of a Science or Arts degrees, could lead students to experience mental health issues such as stress, anxiety and depression which possibly could lead to dropout.

Another form of intervention offered to students in many third level institutions is that of a support centre. The role of a support centre is generally to assist students who are struggling with their coursework (or some aspect of it) without directly covering their current assignment. They aim to be supportive, judgement free and helpful towards students. Support centres such as these have been shown to have a particular benefit to at-risk students (Mac an Bhaird, Morgan, & O'Shea, 2009).

The prevalence of computer science and programming support centres across third level institutions in Ireland is encouraging, with almost all institutions offering some form of support structure. One of the first such centres in Ireland was that of Trinity College Dublin's Science department (Stamouli, Doyle, & Huggard, 2005). Their initial findings were positive but did not look directly at what effect the centre had on grades. Since 2011, the Department of Computer Science at Maynooth university has offered a support service to students. The original Programming Support Centre (PSC) was set up to provide support for two first year modules. The pedagogical approach of the PSC was originally presented in 2015, which outlined how the PSC was set up in and provided recommendations on such (Nolan, Mooney, & Bergin, 2015). Since then the PSC has undergone radical change and has been rebranded as the "Computer Science Centre" (CSC).

Computer Science support centres often focus most of their resources on CS1 students. This focus on CS1 is largely due to the difficult nature of CS1. CS1 is the first formal module taken in Computer Science by students and given the worrying non-progression rates of CS1 there is a need to try and retain students (Connolly & Murphy, 2005). This is an important time to offer support as retaining of students from first to second year computer science has always been a difficult task (Quille, Bergin, & Mooney, 2015).

This paper will discuss the structure and practices of the Computer Science Centre in our university. In addition, a small study was carried out which will examine the self-reported

self-efficacy levels of Higher Diploma students at several intervals over the duration of a 3-week intensive CS1 course. These Higher Diploma students have previously earned a degree in a non-computing subject. We aim to determine if the reported levels of both change during the course, and if there are any notable differences between students who attend the CSC and those who don't. From there, determinations can be made about what interventions can be offered to students who are experiencing mental health or confidence issues.

## **2. Background and Related Work**

Computer science in Ireland has become a state examined subject in 2020 meaning that some students entering third level in the future will have a formal exposure to computer science. However, the majority currently in third level don't have any formal exposure to CS. Going forward the provision of CS in second level is on a pilot basis and uptake within schools may be slow. This can present an issue in third level where students are taking modules in CS without the basic skills to cope with the demands being placed on them. This is a similar issue to the "Maths problem" which is not just an issue for students undertaking mathematics courses, but it is also considered an issue for students in science, technology and engineering, and has been shown to be a contributing factor to the low retention rates in these disciplines (Berry, Mac An Bhaird, & O'Shea, 2015). In response to the "Maths problem" the provision of Mathematics Learning Support is a regular addition in many HEIs worldwide.

Such Computer Science Learning Supports are not as commonplace, but they are appearing in more HEIs. Computer Science Centres are additional services to students on top of their lectures, tutorials and labs, and are never meant as a substitution for these. Students are always encouraged to engage fully with the course and carry out independent work before visiting a centre. Much of the work within these centres is on remedial supports but some centres support all students, irrespective of their performance in the subject, by providing advanced supports for stronger students.

### ***2.1. Self-efficacy in Computer Science***

The concept of self-efficacy has seen an increase in interest over the past 10 years. Conducting a search on the ACM Digital Library for "self-efficacy" returns 111,575 hits with the earliest being in 1954. Filtering this search down to since 2010 returns 73,458 and further filtering to 2018 returns 20,744 hits [Correct as of 01/04/2020].

Programming self-efficacy has been a topic of increased research in the area of computer science and particularly amongst first-year computer science students. Bandura defined perceived self-efficacy as "*people's judgments of their capabilities to organize and execute courses of action required to attain designated types of performances*" (Bandura, 1986). Self-efficacy mediates between an individual's knowledge and their actions. Therefore, somebody

may possess the knowledge or skills which are necessary to perform a particular task, however, they may not succeed due to their self-doubt or self-belief in their ability. Self-efficacy theory has emerged as an important means of understanding and predicting a person's performance.

Bergin showed that student's belief in their programming ability was the highest factor in predicting programming success (Bergin & Reilly, 2005, 2006). Quille et al., in a re-validation study found that programming self-efficacy was again found to be the main predictor of success in first-year computer science (Quille, Culligan, & Bergin, 2017). Research has shown that those with high programming self-efficacy are more likely to undertake tasks that are more challenging and demonstrate a want to learn and engage with material (Schunk, 1989). In addition to this, the success of a task will most likely increase the programming self-efficacy of the person involved (Schunk, 1989). In contrast, those with low programming self-efficacy tend to experience tasks that are easy or at least easier, as much more difficult than they are. This leads to the person experiencing stress or anxiety (Bandura, 1986). This would lead to the student not succeeding and further reduce their programming self-efficacy.

### **3. Computer Science Centre**

The pedagogical approach at the CSC is based on peer-tutoring and at the heart of the model is the volunteer nature of this tutoring. There is no payment or credits provided for tutors, rather these are second- and third-year students who give their time freely, usually because they would like to experience this type of role or have a genuine desire in helping others. The tutors are proficient in the two introductory modules. The idea is that, having just recently finished the modules themselves, they are the closest to the material, and to the current student's age to have an effect on how they absorb the material. These tutors are trained in the first week of semester one, with a refresher session provided in semester two. Training involves guiding the tutors on how to talk to (and more importantly listen to) students, how to use the CSC systems and online support structures, and how to use good teaching methodologies to instil confidence in the students. Expectations of the tutors while working are also covered.

This approach, while sharing many philosophies with the Peer Assisted Study Session (PASS) (Geerlings, Cole, Batt, & Martin-Lynch, 2016) paradigm in promoting improved student comprehension, performance and retention, is different in that the number of students being tutored by an individual tutor at any time is small and these students may be working on different topics. Similarly, to PASS students are encouraged to work together in the CSC but it is not required, thereby allowing differentiation in student learning.

As the CSC is reliant on volunteers, there is a limited number of hours that the CSC is open to students. These hours are usually dictated by when 1) the students of first and second year can attend and 2) the volunteer tutors can attend. As such, the CSC is open for between 10-15 hours a week for drop-in services.

The CSC is coordinated by two academic staff tutors. Each session is managed by at least one of the tutors. Lecturers do not attend the CSC, a deliberate decision, to encourage a relaxed setting for learners.

Given the setup of the CSC, learners are enticed into the centre through the various services that are offered. With its re-branding the CSC offers additional supports such as; a) provide an induction session with new students, b) provide physical self-help sheets for all supporting modules, c) provide weekly programming competitions for both first- and second-year CS students and d) provide support for extra-curricular activities such as building robots and coding hardware boards such as Arduino, Raspberry PI and MicroBits.

If for some reason students cannot make one of the drop-in hours, the CSC tutors are available in an online environment through technologies such as online ticketing systems, social media technologies and Slack channels. Having this online presence, particularly through social media has increased the profile of the centre and has turned the centre from a traditional support service to more of a social setting where students collaborate on different CS topics.

The CSC has been in operation since the 2011 with 3 of the founding members still involved in the running of the centre today. While no formal studies have been carried out in the centre, reports from the participating students indicate that the CSC has improved their programming ability which, in turn, has afforded them higher marks in both continuous assessments and final module exams. In the first semester of the 2019-2020 academic year, 49% of first-year students attended the CSC at least once in the semester, with 26% of these students attending at least once every week. The average CS1 exam mark of those who attended the CSC was 55.22%, compared to 49.95% for those who did not attend.

## **4. Early Indications**

Given that no formal studies have been carried out on the CSC to date, this paper attempts to investigate if the centre has improved the programming self-efficacy of attending students.

### **4.1. Instruments**

As part of this experiment, we included Bergin's Programming Self-efficacy questionnaire. The scale consists of 10 questions and has been shown to have generally inter-item and test-retest reliability (Bergin & Reilly, 2005, 2006). Each of the questions relate to programming

self-efficacy. For example, the first question is “At times I think that I am no good at all at programming” with responses on a Likert scale, “Not true of me” – “Very true of me”.

#### **4.2. Participants**

The participants who took part in this study were all mature students (> 23 years of age) participating in an intense three-week course in Introduction to Programming taught through Java. The course covers aspects from system output to classes and objects and inheritance. In order to be eligible for the course, all students must retain a degree from a relevant university, however, the degree must not be in the area of computer science or software development.

Everyone in the course had the same lecture contact and lab contact. Attending the CSC was a voluntary choice the students made. The module assessment is carried out over the three weeks of the course. Each day there are two lab assessments. These lab assessments account for 20% of the final mark. On the Friday of each week, there was a written exam and from the first two of these exams, the best result is taken and is worth 40% of the final mark. The final exam on the last Friday is worth 40%. All grades were collected.

#### **4.3. Data Collection**

Microsoft forms were used to collect answers to the questions from the background questionnaire and self-efficacy surveys. The survey was conducted in multiple steps. On Day 1, participants took a demographic survey and the programming self-efficacy questionnaire. On Day 5, Day 8, Day 11 and Day 14 participants took only the programming self-efficacy questionnaire. Full ethical approval was obtained to carry out this research.

### **5. Results**

#### **5.1. Participant Profile**

A total of 39 students participated in all 5 surveys. Table 1 shows the gender and age breakdown for those who attended and didn't attend the CSC.

**Table 1 Gender and Average Age breakdown of all participating students.**

	<b>Gender</b>	<b># Participants</b>	<b>Average Age</b>
Attended	Male	11	31.2
	Female	6	29.6
Did Not Attend	Male	15	27.4
	Female	7	29.57

### 5.2. Programming Self-efficacy

Programming Self-efficacy was captured at five points across the three-weeks. Figure 1 shows the changes of programming self-efficacy in the participants of the CSC, those who didn't attend and the overall average change in programming self-efficacy during the three-week course. Initially, for the participants who didn't attend the CSC their programming self-efficacy was low. Following Survey 2 and Survey 3, there was a sharp increase in participants programming self-efficacy. Seeing this improvement in self-efficacy was positive. However, the sharp increase in self-efficacy is then followed by a decrease. This decrease is almost a self-correction of the student's appraisal of their programming ability. Examining those who did attend the CSC, their programming self-efficacy was low however their programming self-efficacy improved and ultimately finished at the same programming self-efficacy level as those who didn't attend the CSC. Table 2 shows the difference in programming self-efficacy between those who attended the CSC and those who did not.

**Table 2 Difference in programming self-efficacy between those who attended the CSC and those who did not.**

Survey Number	1	2	3	4	5	p-value
Did Not Attend	26.18182	28.81818	29.63636	29.09091	26.27273	0.017224
Attended	23.64706	24.29412	25.17647	24.94118	26.35294	0.722908
p-value	0.227536	0.019668	0.019489	0.022738	0.909608	

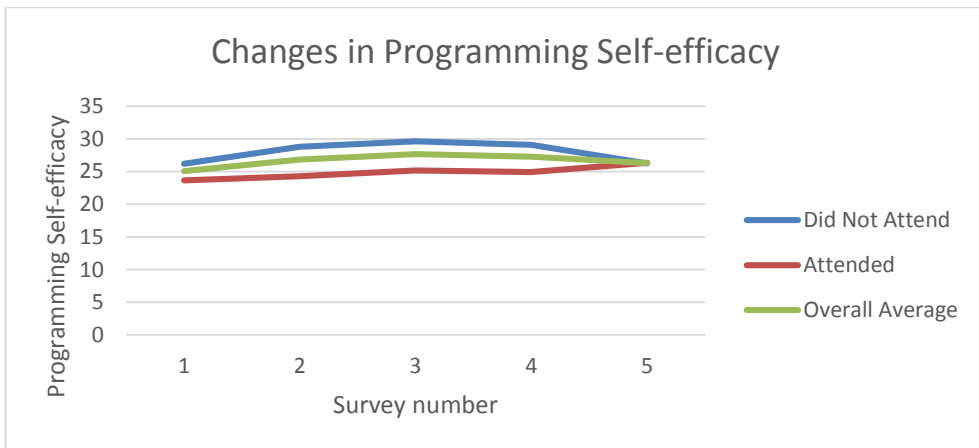


Figure 1. Graph of change of average programming self-efficacy over three weeks.

## 6. Future Work

While there was a decrease in self-efficacy between Survey 3 and Survey 5 overall, this decrease should not be looked on as a negative. The decrease is due to the participants self-correcting their appraisal of their ability. Overall, it appears that the CSC has improved the self-efficacy of the students that attended the CSC over the course of the three weeks. This is the beginning of a formal evaluation of the impact of the CSC and indicatively the CSC is making a positive impact on the students who avail of the services. A formal, longitudinal study is planned to evaluate if the CSC is making a difference to the performance and self-efficacy of the students using it.

## Acknowledgements

The authors would like to thank the participants of this study.

## References

- Bandura, A. (1986). Social foundations of thought and action. *Englewood Cliffs, NJ, 1986*.
- Bergin, S., & Reilly, R. (2005). Programming: Factors That Influence Success. *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*, 411–415. <https://doi.org/10.1145/1047344.1047480>
- Bergin, S., & Reilly, R. (2006). Predicting introductory programming performance: A multi-institutional multivariate study. *Computer Science Education*, 16(4), 303–323. <https://doi.org/10.1080/08993400600997096>
- Berry, E., Mac An Bhaird, C., & O'Shea, A. (2015). Investigating relationships between the usage of Mathematics Learning Support and performance of at-risk students. *Teaching Mathematics and Its Applications: An International Journal of the IMA*, 34(4), 194–204. <https://doi.org/10.1093/teamat/hrv005>
- Bhaird, C. M. A., Morgan, T., & O'Shea, A. (2009). The impact of the mathematics support centre on the grades of first year students at the National University of Ireland Maynooth. *Teaching Mathematics and Its Applications*, 28(3), 117–122. <https://doi.org/10.1093/teamat/hrp014>
- Connolly, C., & Murphy, E. (2005). Retention initiatives for ICT based courses. *Proceedings Frontiers in Education 35th Annual Conference*, S2C-10. <https://doi.org/10.1109/FIE.2005.1612215>
- Geerlings, P., Cole, H., Batt, S., & Martin-Lynch, P. (2016). Peer Assisted Study Sessions (PASS): Does Gender Matter?. *Journal of Peer Learning*, 9, 10–25.
- Harmon, D., & Erskine, S. (2017, December). *Eurostudent Survey VI*. Retrieved from <http://hea.ie/assets/uploads/2018/01/HEA-Eurostudent-Survey.pdf>
- Nolan, K., Mooney, A., & Bergin, S. (2015). Facilitating student learning in Computer Science: large class sizes and interventions. *International Conference on Engaging Pedagogy*.



- Quille, K., Bergin, S., & Mooney, A. (2015). PreSS#, A Web-Based Educational System to Predict Programming Performance. *International Journal of Computer Science and Software Engineering (IJCSSE)*, 4(7), 178–189.
- Quille, K., Culligan, N., & Bergin, S. (2017). Insights on Gender Differences in CS1: A Multi-institutional, Multi-variate Study. *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education*, 263–268. <https://doi.org/10.1145/3059009.3059048>
- Schunk, D. H. (1989). Self-efficacy and achievement behaviors. *Educational Psychology Review*, 1(3), 173–208.
- Stamouli, I., Doyle, E., & Huggard, M. (2005). Establishing structured support for programming students. *34th Annual Frontiers in Education, 2004. FIE 2004.*, 679–683. <https://doi.org/10.1109/fie.2004.1408612>