INFORMATION SYSTEMS EDUCATION JOURNAL

Volume 22, No. 1
March 2024
ISSN: 1545-679X

In this issue:

iscap
Information Systems & Computing Academic Professionals

The **Information Systems Education Journal** (ISEDJ) is a double-blind peer-reviewed academic journal published by **ISCAP** (Information Systems and Computing Academic Professionals). Publishing frequency is five times per year. The first year of publication was 2003.

ISEDJ is published online (https://isedj.org). Our sister publication, the Proceedings of the ISCAP Conference (https://iscap.us/proceedings) features all papers, abstracts, panels, workshops, and presentations from the conference.

The journal acceptance review process involves a minimum of three double-blind peer reviews, where both the reviewer is not aware of the identities of the authors and the authors are not aware of the identities of the reviewers. The initial reviews happen before the ISCAP conference. All papers, whether award-winners or not, are invited to resubmit for journal consideration after applying feedback from the Conference presentation. Award winning papers are assured of a publication slot; however, all re-submitted papers including award winners are subjected to a second round of three blind peer reviews to improve quality and make final accept/reject decisions. Those papers that are deemed of sufficient quality are accepted for publication in the ISEDJ journal. Currently the target acceptance rate for the journal is under 35%.

Information Systems Education Journal is pleased to be listed in the Cabell's Directory of Publishing Opportunities in Educational Technology and Library Science, in both the electronic and printed editions. Questions should be addressed to the editor at editor@isedj.org or the publisher at publisher@isedj.org. Special thanks to volunteer members of ISCAP who perform the editorial and review processes for ISEDJ.

INFORMATION SYSTEMS EDUCATION JOURNAL

Editor Comment:

We are pleased to present our first issue of 2024, including papers on AI in and out of the classroom. We are also delighted to welcome Dr. Kevin Mentzer to the editorial team this year, joining Dr. Dana Schwieger as Associate Editor, along with Drs. Ira Goldstein and Michelle Louch as Cases & Exercises Co-Editors. Thank you to each of these individuals, to our reviewers, and to our Publisher, Dr. Tom Janicki.

Paul Witman, Senior Editor, ISEDJ

Editors

Student-Driven Programming Instruction: A Follow-Up Study

Deepak Dawar
daward@miamioh.edu
Dept. of Computer and Information Technology
Miami University
Hamilton, Ohio 45011, U.S.A.

Abstract

Learning computer programming is typically difficult for newcomers. Demotivation and learned helplessness have received much attention. Besides the subject's intricacy, low in-class participation has been associated with poor student achievement. This paper presents a follow-up, stage 2 study on the novel instructional technique, Student-Driven Probe Instruction (SDPI), to address low in-class participation in programming courses. Instead of the teacher lecturing/explaining content to the class and asking questions, students were shown a snippet of code or other relevant material and given the option to ask questions beforehand. The study was conducted in two stages: stage 1 pilot and stage 2. This paper presents the results of stage 2, while stage 1 operations and results are discussed briefly. The number of questions asked in class, real-time Trello board postings, and emails/Slack conversations with the instructor were used to track participation. In-class participation showed significant improvement. Average quiz and in-class activity scores showed marginal gains. Results from the end-of-course survey show that students preferred SDPI over the traditional lecture style since it stirred their interest in the content and provided them the confidence to ask questions in class. The study is purely exploratory in nature, and no conclusions can be drawn due to the extremely small sample size of the student population.

Keywords: Class participation, introductory programming, pedagogy, student demotivation.

Recommended Citation: Dawar, D., (2024). Student-Driven Programming Instruction: A Follow-Up Study**.** *Information Systems Education Journal,* 22(1), pp.14-29. https://doi.org/10.62273/ UQWB4192

Student-Driven Programming Instruction: A Follow-Up Study

1. INTRODUCTION

The majority of novices find learning computer programming challenging, and significant failure rates are frequently recorded (Allan & Kolesar, 1997; Beaubouef & Mason, 2005; Bennedsen & Caspersen, 2007; Howles, 2009; Kinnunen & Malmi, 2006; Mendes et al., 2012; Newman, Gatward, & Poppleton, 1970; Sheard & Hagan, 1998; Watson & Li, 2014). Due to the difficulty of the material, many students lose interest (Kim & Lerch, 1997; Rogalski & Samurçay, 1990; Robins, Rountree & Rountree, 2003). A large body of data demonstrates that class engagement enhances student performance and significantly affects GPA (Credé, Roch, & Kieszczynka, 2010). No matter how the course is delivered—synchronously or asynchronously—many studies have shown that active class involvement significantly improves student outcomes (Duncan et al., 2012; Nieuwoudt, 2020).

The author has noticed that only a small percentage of the students tend to ask the majority of the questions after having taught numerous programming courses over a period of several years. A few of the others occasionally engage in conversation, but the majority mostly just observe. This is supported by research done by Bowers in 1986 as well as the data acquired for this examination.

Many approaches have been proposed to address this low level of involvement. Although there is a lot of research on these techniques, pair programming (Dongo et al., 2016; Williams et al., 2002) and gamification (Beavis, 2010; Majuri et al., 2018; Osatuyi et al., 2018; Seaborn & Fels, 2015) are some of the most popular ones. These techniques work well and have been found to have a neutral to positive impact on students' academic results.

The author has employed these techniques, and it appears that they have improved the degree of participation in the class's problem-solving activities. One aspect that these methods had little impact on was the students' comfort level with asking questions. For instance, the author has repeatedly observed that many students return to their usual habit of non-participation when it comes to asking questions after a pair programming or gamification session. Most of the questions were asked by the same students who had been asking them earlier, both before and after these exercises.

Students have been encouraged to ask questions in class using various strategies. The two important ones are giving weight to in-class questions (Berdine, 1986; Smith, 1992) and the Random Selector Model (Allred & Swenson, 2006). Assigning points to students who ask questions is an effective example of an external motivator since it gives them a reason to do so. The achievement or curiosity of the students may or may not be impacted by this. Similarly, cold calling boosts engagement while simultaneously raising the class's stress levels (Moguel, 2004).

One question must be addressed in light of the prevalence of low in-class participation rates: why don't certain students participate or raise questions in class?

Regardless of the physical characteristics of the classroom, a student's own anxieties about coming across as insufficient or incompetent in front of others may also prevent them from participating in class (Fritschner, 2000; Hyde & Ruth, 2002; Weaver & Qi, 2005), especially when it comes to raising questions. Students also stated (Armstrong & Boud, 1983; Wade, 1994) that they were most deterred from asking questions due to their lack of confidence. Because they are worried about what other students would think of them, many students choose not to participate (Fritschner, 2000).

Many students may decide not to participate in class due to the heavy conceptual load (Sweller, 1988, 1994) of computer programming because it is challenging to comprehend multiple concepts at once. Many students begin to believe they cannot excel when this happens frequently. According to Crego et al. (2016), this is referred to as "acquired helplessness." This can lead to even lower confidence levels for students.

Student-Driven Probe Instruction (SDPI) intervention was designed with these considerations in mind. The concept was straightforward but counterintuitive: students were presented with a piece of code or content instead of the instructor leading the class by explaining the material and encouraging questions. The instructor then opened the floor to questions without offering any clarifications. The intention was threefold:

1. Lessen the initial mental strain and allow the students to interpret the information themselves first. As a result, rather than being viewed as something the student must be assessed on, the connection between the student and the subject is purely exploratory.
2. Allow the students' questions to guide the lesson rather than the teacher's. Giving the students greater control through this modification might increase their self-esteem and aid them in letting go of their feelings of inadequacy.
3. Anonymize the questions, i.e., the students who do not want to be identified as having asked a particular question but intend to participate should be afforded that opportunity. This is achieved through anonymous cards in Trello boards and is discussed in section 2. This feature was added during stage 2 based on the student feedback from stage 1.

The technique rests on three central pillars, as shown in Figure 1. It is essential that the instructor maintain a non-judgmental classroom environment during this process, as students are now going to ask the most rudimentary questions. Judging their questions as lofty and not relevant will derail the whole process immediately.

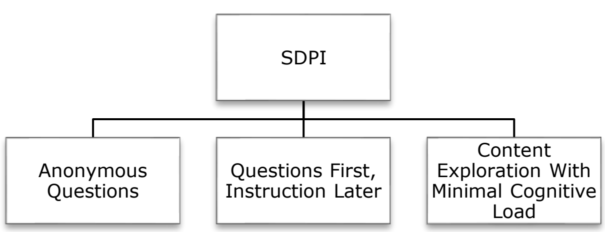

Figure 1: Student-Driven Probe Instructional Technique

Two research questions were addressed in this study:

a) How does the SDPI method affect students' participation in class?
b) If any, how does SDPI affect student grades?

The terms participation and questions asked are interchangeable in this study. The remainder of the paper is organized as follows. The operational features of the approach are discussed in Section 2, along with an explanation of its components. The preliminary findings are presented in Section 3. Section 4 talks about SDPI's advantages and disadvantages. Section 5, which concludes the paper, also briefly outlines the framework for additional research.

2. METHODOLOGY

The study was conducted in two stages.

Stage 1 - Pilot	Single student population P1(12)	
	Pre-mid-term taught with conventional methods.	Post-mid-term taught with SDPI.
Stage 2 - Controlled	Two different student populations Control(13) and Experimental(21)	
	The control group was taught with conventional methods used by the author	The experimental group taught with SDPI

Figure 2: SDPI stages

Stage 1: This stage was done as a pilot project on a single class of the programming course. The author decided it would be too risky to present the SDPI at the start of the course due to its unproven character. The course was divided into two halves for this stage of study. The students were instructed using the traditional method(s) in the first half, during which the topic was taught and student questions solicited. In the second half, SDPI was presented, and the students were just shown a small portion of the code without any explanations. The strategy can be encapsulated as follows:

1. Students will be shown a piece of code/content at the beginning of the class.
2. A certain amount of time is given to the students, generally two minutes, to come up with questions about the content if they have any.
3. It is presumed that all students fully understand the subject matter if there are no queries from the class. The teacher selects a student at random and inquires about the subject matter to test this assumption. This step in the process is crucial because it teaches the students that it is preferable to ask questions than to wait for the instructor to ask them and risk being unable to respond.

4. As the queries come in, they are noted as comments on the source code for later use.
5. The instruction starts and is modeled around the questions whenever enough questions have been asked (often 5–10). The questions are now used as a tool to examine and explain the material.

A sample load is presented below to describe the procedure effectively.

```java
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;
//Q1. What is File?
//Q2. Is the name of the file input.txt or file itself?
//Q3. Where is input.txt stored?
//Q4. Will the Scanner run as long as there is some input in the file?
//Q5. Will the loop stop after the last line is printed?
//Q6. What does throws clause do?
//Q7. Why file in Scanner not the file name?
//Q8.
//Q9.
public class Files {
   public static void main(String[] args){
     try{
       File file = new File("input.txt");
       Scanner in = new Scanner(file);
       while(in.hasNext()){
         String temp = in.next();
       }
     }catch(FileNotFoundException e){
       System.out.println("File Not Found");
     }
   }
}
```

Figure 3: Sample snippet for SDPI stage 1

Figure 3 shows the opening snippet of a particular module. Only after a reasonable number of inquiries about the snippet—in this example, seven—had been made could the instruction start. As soon as a student began to ask a question, it was noted on the source file, which was then shared with the class later.

It is to be noted that stage 1 was a pilot to ascertain whether SDPI should be pursued for further study in a controlled environment. The results were encouraging as class participation rose significantly, and quiz scores improved meaningfully (Dawar, 2022). It is to be noted that if this stage had not produced any meaningful student outcomes, conducting the subsequent stage two study would have been irrelevant.

Two main shortcomings were identified during Stage 1.

1. The first challenge of SDPI stage 1 was in-class data collection. The author had to record (in a matrix) the number of questions asked by each student during every class. This occasionally caused the author to become slightly distracted because they had to simultaneously mark the question in the matrix and write it on the source file so that everyone could see it on a shared computer screen.
2. Some students who did not participate revealed in the after-course interview that they wanted to ask questions but were not comfortable being identified as having asked those questions.

These issues were addressed in stage 2.

Stage 2: This stage study spanned two semesters and involved two sets of students taking the same programming course. These students were divided into control and experimental groups. Students in the control group were taught with the conventional method that the author employs, i.e., the content is explained, and the questions are solicited from students afterward. Students in the experimental group were taught using SDPI, with some of the shortcomings of stage 1 addressed.

One of the major feedback from the stage 1 pilot was that many students did not want to be identified while asking questions, but they wanted to participate. Every student was assigned an anonymous ID on the Trello Board to address this apprehension, such as S1 and S2. During the class, students would type their questions anonymously under their IDs. This would serve a dual purpose – let the intending students participate anonymously while serving as a record of the questions asked by the whole class as well as taking pressure off from the instructor for writing all the questions themselves. A sample is shown in Figure 4. It shows three students with assigned ID's as S1, S2, and S3. These students do not know what ID belongs to whom, thus anonymizing the questions.

Student Population

Despite the lack of clear definitions in the literature, our department's student body includes both traditional and non-traditional students. For the sake of this work, the author defines "traditional" as full-time students who have just graduated from high school. Non-traditional students include those who work full-time, attend school part-time, are older, and are looking for a new career, among other situations.

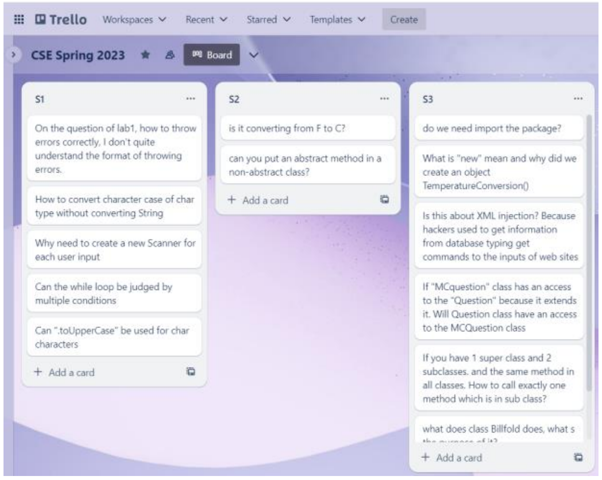

Figure 4: Anonymous Trello Board

The number of students in the stage 1 pilot group was 12. Stage two had 13 students in the control group and 21 students in the experimental group who participated in the study. Students majoring in information technology (IT) can take the course as an elective, but computer science (CSE) students are required to take it.

Data Collected
Numerous factors have been used to evaluate student achievement. Course grades, term grades, and cumulative GPA are the most often utilized metrics (Teixeira, 2016). Student ratings and pre and post-test scores are occasionally used (Omar, Bhutta, & Kalulu, 2009; Felisoni & Godoi, 2018).

For this work, the following data were collected for each participating student:
1. No. of questions asked in class
2. No. of email/Slack contacts with the instructor
3. Quiz scores
4. In-class activity scores
5. Mid-term and final exam scores

3. RESULTS

Findings from the study can be categorized into two groups: quantitative data analysis to examine the potential impacts of SDPI on student engagement and outcomes and student impression of SDPI as revealed by an end-of-course survey.

During the class, each student's total number of questions was kept track of both in the control and experimental groups. The average number of questions asked in class by all participating students is shown in Table 1. In the control or traditional mode of instruction, 65% of the questions were asked by the top 30% of the students. The bottom 45% asked only 15% of the total questions asked by all students, i.e., 45% of the students contributed to only 15% of all the questions asked in the control group. This skew perfectly captures the low participation rates among certain students. This is also consistent with the authors' experience teaching computer programming over the years. The average number of questions asked by the control group was 1.63 per student, while for the experimental group, it was 3.44.

Student No	Average No. of Questions Asked by Each Student in the Control Group	Average No. of Questions Asked by Each Student in the Experimental Group
1	0.0	3.0
2	4.5	8.75
3	3.5	3.25
4	1.25	5.0
5	3.75	5.75
6	0.5	4.0
7	0.25	5.25
8	2.0	5.0
9	2.25	5.75
10	0.75	2.25
11	0.0	3.25
12	2.5	6.75
13	1.5	2.0
14	-	3.25
15	-	1.0
16	-	2.75
17	-	4.25
18	-	3.75
19	-	2.0
20	-	0.5
21	-	4.25
Average	**1.63**	**3.44**

Table 1: Average no. of questions asked by each student in class

As evaluated by the number of questions posed by each student, class involvement dramatically increased with the implementation of SDPI. The average number of questions raised in class rose significantly. This is a significant advancement. Taking a closer look at the table prompts the following inquiries:
1. Why did the number of questions asked increase?
2. Was the increase uniformly distributed among students?

The students had to ask questions for the session to move forward and get the content taught because it wasn't explained. This is unquestionably one of the causes of the sharp increase in inquiries.
In the experimental group, 70% of the questions were asked by 66% of the students. This is a much better distribution than the control group, where 65% of the questions were asked by 30%

of the students. This clearly shows the uniform participation among students achieved with SDPI. This is a significant and advantageous development. This is further supported by data from a course evaluation survey, where most students reported that SDPI increased their level of participation in class. An analysis of variance (ANOVA) test was done for this data. The difference was found to be significant, with a p-value of 0.001. This shows that statistically speaking, participation did increase dramatically. For a detailed ANOVA report, see Appendix C.

Table 2 shows the average instructor contacts by students in the control and experimental groups. These are the follow-up, and sometimes new questions and queries students pose outside of class time. The utility of this metric is to gauge the interest in the content outside the classroom. These are collected because, no matter what, some students prefer asking questions one-on-one with the instructor.

Student No	Average No. of Instructor Contact – Control Group	Average No. Instructor Contact – Experimental Group
1	2	3
2	2.25	2.25
3	1	2.25
4	3.25	4
5	2	2.75
6	1.25	3
7	1.5	1.25
8	1	3
9	1.5	1.75
10	1.5	2
11	0	2.25
12	2	2.75
13	1	0.5
14	-	1.25
15	-	2
16	-	2.25
17	-	3.25
18	-	2.75
19	-	2
20	-	0
21	-	1.25
Average	**1.55**	**2.16**

Table 2: Average instructor contacts by students

Contacts made via email and Slack, a team collaboration tool, were considered. The average contacts for the experimental group increased from 1.55 to 2.33. This means that, outside the class and on average, each student contacted the instructor 1.55 times in the control compared to 2.16 times in the experimental group. This is a significant improvement (a p-value of 0.06), though this improvement also comes with a substantial load for the instructor.

Quiz No.	Average Quiz Scores for Control Group	Average Quiz Scores for Experimental Group
1	19.56	23.25
2	31.2	31.91
3	19.68	21.44
4	35.43	36.82
5	28.06	28.2
Average	**26.78**	**28.34**

Table 3: Average quiz scores

Class Activity No.	Average Activity Scores for Control Group	Average Activity Scores for Experimental Group
1	9.375	9.78
2	18.625	18.8
3	16.866667	19.42
4	18.25	19.87
5	28.4375	28.43
6	12.25	12.9
7	15.5	15.7
8	12.25	15.16
9	32.25	31.45
10	17.4	18.4
Average	**18.10**	**19.01**

Table 4: Average in-class activity scores

Tables 3 and 4 present the average quiz and in-class activity scores obtained by the control and experimental groups. There were five quizzes and ten in-class activities in total, with different points depending on their complexity. Both groups were administered the same quizzes and activities. This is only a marginal improvement in these scores. Hence, this is a mixed yet positive result.

Table 5 shows a comparison between average exam scores for both groups. The final exam was worth 100 points, and the midterm was worth 50. No conclusion can be drawn at this point regarding the impact of SDPI on exam scores. More iterations of SDPI need to be run to see if these results hold or improve.

Group	Control	Experimental
Mid-term Exam Average Score (50)	38.75	44.56
Final Exam Average Score (100)	84.1	83.5

Table 5: Exam scores

End of Course Survey

Regarding SDPI, a final anonymous survey was conducted for the experimental group. Table 6 lists a few survey questions (the whole survey is attached in Appendix B).

Question	Definitely Yes	Probably Yes	Might or Might Not	Probably Not	Definitely Not
1.Made you more participative	53%	14%	4%	23%	4%
2.Improved understanding of material	48%	28%	4%	19%	4%
3.Made you curious about the content	43%	28%	14%	9%	4%
4.Made you pay attention to the material	43%	28%	19%	5%	5%
5.Made you feel confident about asking questions	43%	19%	28%	9%	0%

Table 6: End-of-course survey responses

Nearly 68% of all the students who filled out the survey said that SDPI increased their participation. This is very encouraging, and in line with the results of Stage 1. 76% of the students said that their understanding of the subject had probably increased with the use of SDPI, whereas 19% reported no change.

According to Kidd and Hayden (2015) and Szumowska and Kruglanski (2020), curiosity is a learning catalyst. It is encouraging to see that 71% of the students said that SDPI increased their curiosity about the subject matter. Most students said that SDPI sharpened their attention to the lecture subject.

An important question on their level of stress while using SDPI was posed to the class. Given that a stressful learning environment may result in demotivation and lower learning effectiveness (Bowers, 1986), this was one of the survey's most crucial questions. If having to ask questions in class made students anxious, SDPI would fall short right away. Interestingly, 53% of respondents stated that SDPI decreased their stress levels, 24% said it had no effect, and 23% indicated it had increased their stress levels in the class. The practical constraints of any new intervention can be accommodated by this distribution. It is still necessary to look into the root causes of the higher stress that 23% of the students report experiencing.

Additionally, the students were asked about their preferred mode of instruction between traditional and SDPI. 72% of the students preferred the SDPI method, 24% chose the conventional method, and 4% had no opinion. This is encouraging news for the investigation's future and, in the author's view, a little vote of confidence in SPDI.

4. DISCUSSION

Given the small sample size, it is still too early to generalize the technique's efficacy, but the early results provide some fascinating insights.

Strengths

According to classroom and assessment data and student survey responses, most students found SDPI beneficial even though they thought the method was counterintuitive. This is seen by the considerable increase in class participation with SDPI. The author would like to propose that reducing the inadequacy factor among non-participating students is one cause of this development. Because they are worried that other students may judge their inquiries, many students choose not to engage. SDPI mitigates this factor, as the content isn't explained to begin with, and by introducing anonymous questions. Many students hesitate to raise questions after the instructor introduces a specific idea because they believe their inquiries might be perceived as silly. They want to ask questions but do not want to be identified. Asking anonymous questions in class using Trello boards gives them a pathway to participate. The author believes that SDPI provides students with a broad and open range of inquiries without making them feel inadequate.

Additionally, 79% (combined for both stages) of the participants indicated that they would prefer SDPI over a traditional setting. This clearly shows that students are eager to ask questions given a chance and the right environment, and SDPI offers just that.

Challenges

Getting the students used to the idea that their questions, not the instructor's, will determine the direction of the session is very counterintuitive. During both stage 1 and stage 2, the author struggled for a couple of classes to get everyone on board. In standard classroom settings, students are used to the content being explained first and wait for the instructor to take their questions. In a conventional lecture context, the instructor has the majority of the control, and the students are aware of this mechanism. However, with SDPI, a portion of that power is delegated to the students to create their own questions and guide the lesson in a particular direction. It will likely take some time for students to adjust to this change of power.

The fact that the current version of SDPI lacks a way to evaluate the caliber of student inquiries is another major problem. A question about the feasibility/optimality of a code fragment is considered in the same way as a straightforward query about a symbol in the source code. This is a significant flaw in SDPI as it now exists. Future editions of SDPI will include a weighting system that will divide student questions into groups according to the level of complexity they represent.

Time management in class and how thoroughly the content is covered are two other problems. The author often knows how much content will be covered during the class session because they prepare their lectures in advance. Because the instruction was dependent on the student's questions, it was challenging to cover the targeted topic. The questions consumed time that could have been spent on other topics that day because of their vast breadth. As the author takes a few more classes with SDPI, this problem might be lessened.

5. CONCLUSION AND FUTURE WORK

The results of utilizing SDPI in a controlled setting, an experimental teaching method, were presented in this paper. The goal was to compare the participation rates and student outcomes between traditional teaching and SDPI-based settings. Anecdotally, the results suggest that utilizing SDPI may increase student participation in class. A significant improvement in class participation as measured by no. of questions asked was reported. Additionally, a marginal improvement in the average quiz and in-class activity scores was observed after the introduction of SDPI. The mid-term and final exam scores did not have any observable change.

Even though marginal improvements are reported in the experiment, it is emphasized that no formal conclusions can be drawn at this stage due to the very small sample size of the student population. Hence, it would be premature to consider the SDPI approach as a workable tactic for affecting student grade outcomes at this point. However, the preliminary findings are positive, and if used in conjunction with other methods, SPDI offers a clear path for further study.

6. REFERENCES

Allan, V. H. & Kolesar, M. V. (1997). Teaching computer science: a problem solving approach that works. ACM SIGCUE Outlook, 25(1–2),2–10. https://doi.org/10.1145/274375.274376

Allred, C. & Swenson, M. (2006) Using Technology to Increase Student Preparation for and Participation in Marketing Courses: The Random Selector Model, Marketing Education Review, 16 (1), 15-21. https://doi.org/10.1080/10528008.2006.11488932

Armstrong, M., & Boud, D. (1983). Assessing participation in discussion: An exploration of the issues. Studies in Higher Education, 8, 33-44. https://doi.org/10.1080/03075078312331379101

Beaubouef, T. B. & J. Mason (2005). Why the High Attrition Rate for Computer Science Students: Some Thoughts and Observations. Inroads – The SIGCSE Bulletin, 37(2), 103–106. https://doi.org/10.1145/1083431.1083474

Beavis, C. (2010). Literacy, Learning, and Online Games: Challenge and Possibility in the Digital Age. In Proceedings of the IEEE 3rd International Conference on Digital Game and Intelligent Toy Enhanced Learning. Piscataway, NJ: Institute for Electrical and Electronics Engineers. https://doi.org/10.1111/j.1754-8845.2009.01035.x

Bennedsen, J. & Caspersen, M. E. (2007). Failure rates in introductory programming. ACM SIGCSE Bulletin, 39(2), 32–36. https://doi.org/10.1145/1272848.1272879

Berdine, R. (1986). Why some students fail to participate in class. Marketing News, 20, 23-24. https://doi.org/10.13189/ujer.2018.060201

Bowers, J. W. (1986). Classroom communication apprehension: A survey. Communication Education, 35, 372-378.

Credé, M., Roch, S. G., & Kieszczynka, U. M. (2010). Class Attendance in College: A Meta-Analytic Review of the Relationship of Class Attendance With Grades and Student Characteristics. *Review of Educational Research*, *80*(2), 272–295. https://doi.org/10.3102/0034654310362998

Crego, A., Carrillo-Diaz, M., Armfield, J., & Romero, M., (2016). Stress and Academic Performance in Dental Students: The Role of Coping Strategies and Examination-Related Self-Efficacy Journal of Dental Education, 80 (2) 165-172.

Dawar, D (2022). Question Driven Introductory Programming Instruction: A Pilot Study. Journal of Information System Education, 34(2) 231-242.

Dongo, T., Reed, A. H., & O'Hara, M. (2016). Exploring pair programming Benefits for MIS majors. Journal of Information Technology Education: Innovations in Practice, 15, 223-239.

Duncan, K., Kenworthy, A. L., Mcnamara, R., & Kenworthy, D. A. (2012). The Effect of Synchronous and Asynchronous Participation on Performance in Online Accounting Courses. *Accounting Education*, *21*(4), 431–449. https://doi.org/10.1080/09639284.2012.673387

Felisoni, D. D., & Godoi, A. S. (2018). Cell phone usage and academic performance: An experiment. Computers & Education, 117, 175–187. https://doi.org/10.1016/j.compedu.2017.10.006

Fritschner, L. M. (2000). Inside the undergraduate college classroom: Faculty and students differ on the meaning of student participation. The Journal of Higher Education, 71, 342-362. https://doi.org/10.2307/2649294

Howles, T. (2009). A study of attrition and the use of student learning communities in the computer science introductory programming sequence. *Computer Science Education*, 19(1), 1–13. https://doi.org/10.1080/08993400902809312

Hyde, C. A., & Ruth, B. J. (2002). Multicultural content and class participation: Do students selfdisclose? Journal of Social Work Education, 38, 241-256. https://doi.org/10.1080/10437797.2002.10779095

Kidd, C., & Hayden, B. (2015). The Psychology and Neuroscience of Curiosity, Neuron, 88(3), 449-460. https://doi.org/10.1016/j.neuron.2015.09.010

Kim, J. & Lerch, F. J. (1997). Why is programming (sometimes) so difficult? Programming as scientific discovery in multiple problem spaces. *Information Systems Research* 8(1) 25–50. https://doi.org/10.1287/isre.8.1.25

Kinnunen, P. & Malmi, L. (2006). Why students drop out CS1 course?. In Proceedings of the Second International Workshop on Computing Education Research (pp. 97–108). New York, NY: ACM. https://doi.org/10.1145/1151588.1151604

Majuri, J., Koivisto, J., and Hamari, J. (2018). "Gamification of education and learning: a review of empirical literature," in Proceedings of the 2nd International GamiFIN Conference, GamiFIN 2018, CEUR-WS, (Finland).

Mendes, A. J., Paquete, L., Cardoso, A. & Gomes, A. (2012). Increasing student commitment in introductory programming learning. In Frontiers in Education Conference (FIE) (pp. 1–6). New York, NY: IEEE. https://doi.org/10.1109/FIE.2012.6462486

Moguel, D. (2004). What does it mean to participate in class?: Integrity and inconsistency in classroom interaction. Journal of Classroom Interaction, 39, 19-29.

Newman, R., Gatward, R. & Poppleton, M. (1970). Paradigms for teaching computer programming in higher education. WIT Transactions on Information and Communication Technologies, 7, 299–305. https://doi.org/10.2495/SEHE940351

Nieuwoudt, J. E. (2020). Investigating synchronous and asynchronous class attendance as predictors of academic success in online education. *Australasian Journal of Educational Technology*, *36*(3), 15–25.

Omar, A., Bhutta, M. K. S., & Kalulu, D. (2009). Assessment of Student Outcomes in Management Information Systems Online Course Participation. 10.

Osatuyi, B., Osatuyi, T., and de la Rosa, R. (2018). Systematic review of gamification research in is education: a multi-method approach. CAIS 42, 95–124. https://doi.org/10.17705/1CAIS.04205

Robins, A. V., Rountree, J. & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education* 13(2) pp. 137–172. https://doi.org/10.1076/csed.13.2.137.14200

Rogalski J. & Samurçay R. (1990). Acquisition of programming knowledge and skills. In J. M. Hoc, T. R. G. Green, R. Samurçay & D. J. Gillmore, eds., Psychology of Programming. London: Academic Press, pp. 157–174.

Seaborn, K., and Fels, D. I. (2015). Gamification in theory and action: a survey. Int. J. Hum. Comput. Stud. 74, 14–31. https://doi.org/10.1016/j.ijhcs.2014.09.006

Sheard, J. & Hagan, D. (1998). Our failing students: a study of a repeat group. ACM

SIGCSE Bulletin, 30(3), 223–227. https://doi.org/10.1145/290320.283550

Smith, D. H. (1992). Encouraging students' participation in large classes: A modest proposal. Teaching Sociology, 20(4), 337-339.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2), 257–285. https://doi.org/10.1207/s15516709cog1202_4

Sweller, J. (1994). Cognitive load theory, learning difficulty, and instructional design. *Learning and Instruction*, 4(4), 295–312. https://doi.org/10.1016/0959-4752(94)90003-5

Szumowska, E., & Kruglanski, A. (2020). Curiosity as end and means, Current Opinion in Behavioral Sciences, 35, 35-39. https://doi.org/10.1016/j.cobeha.2020.06.008

Teixeira, A. A. C. (2016). The impact of class absenteeism on undergraduates' academic performance: Evidence from an elite Economics school in Portugal. Innovations in Education and Teaching International, 53(2), 1–13. https://doi.org/10.1080/14703297.2014.937730

Wade, R. (1994). Teacher education students' views on class discussion: Implications for fostering critical reflection. Teaching and Teacher Education, 10, 231-243. https://doi.org/10.1016/0742-051X(94)90015-9

Watson, C. & Li, F. W. (2014). Failure rates in introductory programming revisited. In Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education (pp. 39–44). New York, NY: ACM. https://doi.org/10.1145/2591708.2591749

Weaver, R. R., & Qi, J. (2005). Classroom organization and participation: College students' perceptions. The Journal of Higher Education, 76, 570-601. https://doi.org/10.1353/jhe.2005.0038

Williams, L., Wiebe, E., Yang, K., Ferzli, M., Miller, C. (2002). In support of pair programming in the introductory computer science course Computer Science Education, 12 (3), pp. 197-212. https://doi.org/10.1076/csed.12.3.197.8618

APPENDIX A

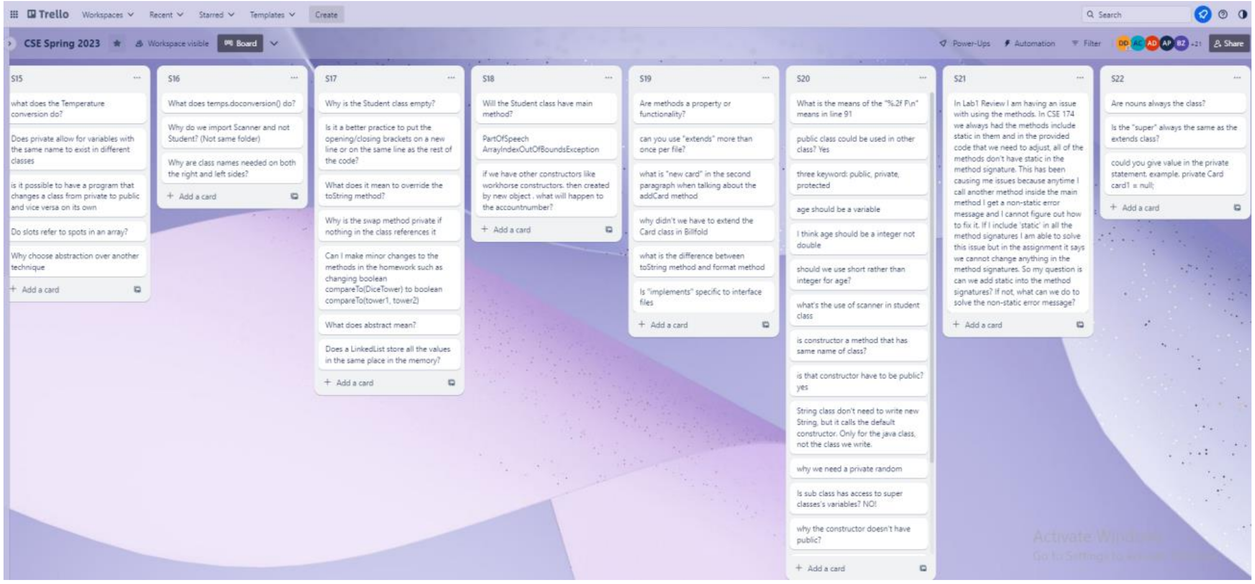

Figure 5: Sample raw data example questions asked by students during the course

APPENDIX B

Survey Instrument for SDPI

Q1 Did the Student-Driven Probe Instructional Approach (SDPI) make you more participative in the class?

- Definitely yes. It made me more participative. (1)
- Probably yes (2)
- Might or might not (3)
- Probably not. I avoided asking questions. (4)
- Definitely not (5)

Q2 During SDPI make you feel confident about asking opening questions?

- Definitely yes. I was confident since I could ask any question about the content. (1)
- Probably yes (2)
- May be (3)
- Probably not. I avoided asking questions. (4)
- Definitely not (5)

Q3 What impact did SDPI have on your stress levels in class?

- It definitely reduced my stress levels. I felt free to ask any type of questions since nothing was explained about the content, to begin with. (1)
- It probably reduced my stress levels. (2)
- It had no impact on my stress levels. (3)
- It increased my stress levels. (4)

Q4 Did the SDPI approach improve your understanding of material?

- Definitely yes. It made me think deeply about the content since I was the one asking the opening questions. (1)
- Probably yes (2)
- Might or might not (3)
- Probably not (4)
- Definitely not (5)

Q5 Did the SDPI approach make you more curious about the content taught in class?

- Definitely yes. By looking at the content that was not explained, I became curious about the content. (1)
- Probably yes (2)
- Might or might not (3)
- Probably not (4)
- Definitely not (5)

Q6 Did the SDPI approach made you pay attention to the material being presented?

- Definitely yes (1)
- Probably yes (2)
- Might or might not (3)
- Probably not (4)
- Definitely not (5)

Q6 Given an option, what mode of instruction would you prefer for this course?

○ The SDPI approach wherein the instructor shows you material, and let you begin asking questions to accommodate everyone's questions and curiosity levels. (1)

○ The traditional approach wherein the instructor explains the content, and then they proceed to ask you questions about the content just explained. (2)

○ No preference (3)

Q7 According to you, what changes should be made to the SDPI format to improve it further?

____

____

____

____

____

End of Block: Questions

APPENDIX C

Analysis of Variance Results

F-statistic value = 11.44715

P-value = 0.00191

Data Summary				
Groups	N	Mean	Std. Dev.	Std. Error
Group 1	13	1.75	1.493	0.4141
Group 2	21	3.8929	1.9535	0.4263

ANOVA Summary					
Source	Degrees of Freedom DF	Sum of Squares SS	Mean Square MS	F-Stat	P-Value
Between Groups	1	36.8712	36.8712	11.4472	0.0019
Within Groups	32	103.0718	3.221		
Total:	33	139.9431			

Figure 6: ANOVA for no. of questions asked in the control and experimental (SDPI) groups

Analysis of Variance Results

F-statistic value = 3.76481

P-value = 0.0612

Data Summary				
Groups	N	Mean	Std. Dev.	Std. Error
Group 1	13	1.5577	0.785	0.2177
Group 2	21	2.1667	0.9465	0.2065

ANOVA Summary					
Source	Degrees of Freedom DF	Sum of Squares SS	Mean Square MS	F-Stat	P-Value
Between Groups	1	2.978	2.978	3.7648	0.0612
Within Groups	32	25.3119	0.791		
Total:	33	28.2899			

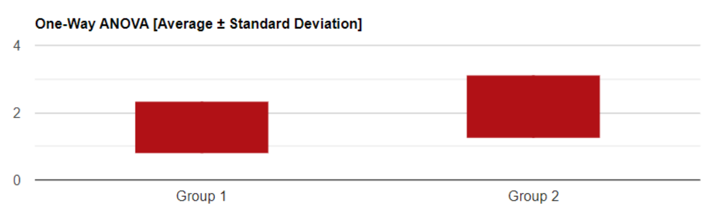

Figure 7: ANOVA for no. of instructor contacts in the control and experimental (SDPI) groups

Analysis of Variance Results

F-statistic value = 0.13262

P-value = 0.72517

Data Summary				
Groups	N	Mean	Std. Dev.	Std. Error
Group 1	5	26.786	7.0451	3.1507
Group 2	5	28.324	6.2887	2.8124

ANOVA Summary					
Source	Degrees of Freedom DF	Sum of Squares SS	Mean Square MS	F-Stat	P-Value
Between Groups	1	5.9136	5.9136	0.1326	0.7252
Within Groups	8	356.7247	44.5906		
Total:	9	362.6383			

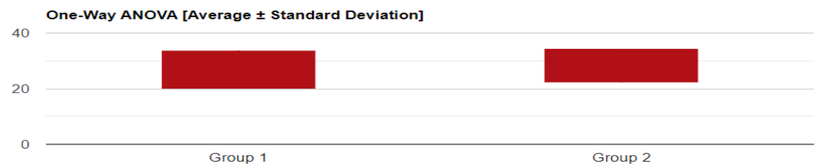

Figure 8: ANOVA for no. average quiz scores obtained in the control and experimental (SDPI) groups