Check for updates

Prior What Experience? The Relationship Between Prior Experience and Student Help-Seeking Beyond CS1

Shao-Heng Ko
shaoheng.ko@duke.edu
Duke University
Durham, North Carolina, USA

Kristin Stephens-Martinez
ksm@cs.duke.edu
Duke University
Durham, North Carolina, USA

Abstract

Background and Context. Prior experience (PE) has been shown to be related to computing students' performance, persistence, and help-seeking behavior. However, most works studied prior *programming* experience in *introductory programming (CS1) courses*, while other forms of PE in other contexts are underexplored.

Objectives. We seek to study the relationship between all kinds of relevant PE and help-seeking behavior in course-affiliated resources within and beyond the typical CS1 context.

Method. We analyzed $N = 2,625$ students' self-reported PE and course-affiliated help-seeking records, originating from 19 offerings of six courses at a post-secondary US institution. Four of the six courses are *non-programming* or *non-introductory* courses.

Findings. We found pronounced *negative* relations between PE and students' *frequency* of seeking one-on-one student-staff interactions, as well as *positive* relations between PE and students' self-reported progress from these interactions. When multiple forms of PE are tracked, the PE that is more aligned with the course activities has a more pronounced effect. We found little relation between PE and students' behavior in class discussion forums.

Implications. Our results highlight that (1) the relationship between PE and help-seeking extends beyond the CS1 context; and (2) this relationship differs between help resources and between course contexts. This motivates providing resource-specific and course-specific training on helping different student populations.

CCS Concepts

• **Social and professional topics** → **Computing education**.

Keywords

Computing Education; Help-Seeking; Prior Experience

ACM Reference Format:
Shao-Heng Ko and Kristin Stephens-Martinez. 2025. Prior What Experience? The Relationship Between Prior Experience and Student Help-Seeking Beyond CS1. In *Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2025), June 27-July 2, 2025, Nijmegen, Netherlands.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3724363.3729092

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

ITiCSE 2025, Nijmegen, Netherlands

© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-1567-9/2025/06
https://doi.org/10.1145/3724363.3729092

1 Introduction

Academic help-seeking is a self-regulated learning strategy that has been linked to successful learning in general [12] and computing education research [27]. Modern post-secondary computing classes offer a rich menu of help resources thanks to the rise of online help delivery [13, 15] and the emergence of generative AI tools [14, 22]. Such an abundance of resources incurs high cognitive load [2], and computing students are now tasked with navigating the complex help landscape with little explicit guidance [21].

The computing education community has studied the effects of prior experience (PE) on students' predicted success [37], actual performance [1, 5, 41, 42], study habits [25], and retention [32] extensively. Regarding help-seeking, students with no/less PE are reported to be less skilled in providing sufficient context when seeking help [41]. They often feel embarrassed, judged, or intimidated around experienced peers [7], potentially due to having unnecessarily high self-expectations [6], which sometimes leads to them hiding their identities [32] when seeking social help.

However, *prior experience* is an overloaded term: it is predominantly used to refer to prior *programming* experience within the context of *introductory programming for CS majors* [25, 32, 37] or a core sequence of CS programming courses [1, 5, 42], while other forms of PE in other computing contexts are less explored. Beyond any computing-related PE, even *prior help-seeking experience in non-computing contexts* has been shown to facilitate the transferral of effective metacognitive strategies to computing contexts [24]. Motivated by these gaps, we seek to answer:

- (RQ) What are the relationships between computing students' *relevant PE* (broadly construed) and course-affiliated help-seeking behavior, both *within* and *beyond* CS1-for-major contexts? Is any relationship context-dependent?

To answer the question, we analyzed students' self-reported relevant PE and help-seeking records in course-affiliated help platforms across 19 offerings of six computing courses at our institution. These six courses include not only introductory programming courses but also non-programming focused and non-introductory courses (see Section 3).

Our results (Section 4) show a pronounced *negative* relationship between PE and students' usage of one-on-one consulting in all but one course, as well as a *positive* relationship between PE and student progress made in these interactions in two other courses. In courses that tracked multiple forms of PE, the PE directly aligned with the activities students do has more pronounced effects on student help-seeking than other forms of PE. We however found no significant relationships between students' PE and their discussion forum behavior, besides a seemingly context-specific effect on students' usage of private posts in an interdisciplinary data science elective.

2 Related Works

PE in computing education. Prior programming experience is known to be positively associated with student performance [1, 5, 25, 41, 42], persistence [1, 26, 32], and self-efficacy [4, 37] at the post-secondary level. PE can be perceived as a form of *perceived social status* [32]: inexperienced students feel embarrassed, judged, or intimidated by experienced peers [7, 23, 37], leading them to underestimate their abilities [6, 23] and become passive in group work [4, 39]. Many institutions thus explored dedicated separate courses for experienced and inexperienced students [17, 42].

However, the majority of this literature focuses on the introductory programming (CS1) context and lack consensus on how PE should be measured [4, 10]. There is no consensus on whether the benefits of PE on student performance persist to higher-level courses [5] or gradually "fade away" [1, 4] as the content becomes less aligned with the typical PE. There is significantly less work exploring relevant PE in non-programming-focused CS contexts (such as discrete math or algorithm design) or interdisciplinary computing contexts such as data science [3].

Help-seeking in computing education. Many help resources have been studied extensively in computing education contexts. For a non-exhaustive list, this includes not only resources affiliated with a course (coined *formal* [12, 20] or *internal* [34]) such as office hours [19, 46] and class forums [32, 33, 41] but also *informal/external* resources such as web search [30, 35] and generative AI [14, 34, 45]. In addition to numerous works on specific resources and their usage cases, there has been a recent line of work studying computing students' help-seeking behavior across all resources [9, 14] or *approach* (what factors into the selection of resources) [20, 21]. These works reveal many entangled factors that influence students' help-seeking decision process, such as whether resources are tied to a designated time/space [9], the effort to phrase a problem [15], and institutional policies [7]; these factors may even create tradeoffs for students.

PE vs. help-seeking in computing education. The relationship between PE and help-seeking is less explored than that between PE and student performance. Some works find no significant differences in students' *frequency* of using internal help resources [19, 46] or their intention to use external resources [34]. However, others report that inexperienced students feel intimidated when seeking social help [7], struggle to effectively seek active help due to the lack of sufficient vocabulary to phrase their problem [41, 44], and remain anonymous on discussion forums more than their experienced peers [32, 33], although it is not clear whether gender (which is often related with PE) is the actual driving factor behind this phenomenon [32, 38]. Beyond any *computing* PE, prior *help-seeking* experience, both positive and negative, has been hypothesized [19, 30, 46] and documented [24] as a potential factor influencing students' help-seeking approaches and behaviors.

Our contribution. Our work is, within our knowledge, the first to quantitatively examine the relationship between student PE and course-affiliated help-seeking behavior in a variety of computing contexts beyond CS1. In choosing this approach, we seek to deepen our understanding on the interplay between PE and help-seeking, and examine whether existing insights extend to higher-level, non-programming, and interdisciplinary courses.

3 Contexts and Methodology

Table 1: Overview of classes and data.

Course	Type	Prerequisite	Sp23	Fa23	Sp24	Fa24	Total
Intro	Entry/Service	-	**159**	**175**	**72**	**56**	462
DSA	Required	Intro or equiv.	238	272	174	129	813
DM	Required	DSA	62		**68**		130
Algo	Required	DSA + DM	202	119	**220**	**94**	635
Data	Elective	DSA/Intro+stats	**159**	**66**	**115**	**62**	402
DB	Elective	DSA + systems		**183**			183

Each class cell shows the number of consenting students who provided some PE information. An empty cell implies the class was not offered or did not collect PE data. Only the bolded classes collected Outcome in the consulting hours app.

Instructional Contexts. Table 1 summarizes our data collection at Duke University, a medium-size, research-oriented private university in the US with 15-week semesters.

Introduction to programming (Intro, taught in Python) is both a service course for non-majors and an entry course to the CS major, with more students reporting *not intending to major* than *intending to major* in CS. Data structures and algorithms (DSA, taught in Java) is the next course in the introductory programming sequence and is required for CS majors. Students with extensive experience (e.g., AP CS) are encouraged to skip Intro and directly take DSA as the entry point to the curriculum.[1]

Discrete math (DM) and algorithm design/analysis (Algo) are two *non-programming focused* required courses for CS majors with either no or minimal programming assignments. These courses form a prerequisite chain together with DSA (see Table 1).

Data science (Data) and database systems (DB) are two popular elective courses. Data has an interdisciplinary student body (58.0% CS majors) and uses Python, although the focus is on applying data analysis packages (such as `pandas` and `scipy`) rather than Python programming. DB is taken almost exclusively by CS majors (96.7%) and emphasizes database systems design with a semester-long full-stack development project.

Throughout this paper, we refer to the six courses above as *courses*, and single offerings of a course as *classes* (19 in total). Our data spans from Spring 2023 (Sp23) to Fall 2024 (Sp24) semesters, although not every course was offered every semester, and not all datasets were collected for every class (see Table 1).

PE Data. Each class asked students to self-assess 1-3 types of relevant PE using four to five predetermined ordinal options (see Table 2 for a summary), although the survey was not always administered right at the beginning of the semester. Throughout the paper, we use the term *variable* to refer to PE data, and the term *metric* to refer to help-seeking records.

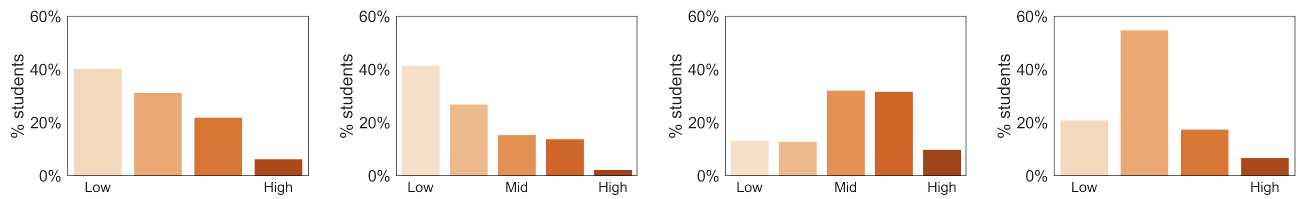
Figures 1-2 illustrate the distributions of all ordinal PE variables *by course*. The Sp24-Fa24 offerings of Data tracked three PE variables. We plotted Data's joint distribution of PE-prog. and PE-Python (across all four offerings) in Figure 2c as they are significantly rank-correlated (Spearman's $\rho = 0.53$, $p = 4.3 \times 10^{-31}$). We plotted # stats courses separately in Figure 1d as it is not significantly rank-correlated with either of the other two variables.

[1] As such, we refrain from calling these courses "CS1"/"CS2", which would wrongly indicate they are the first and second course most students take, respectively.

Table 2: Summary of ordinal PE variables.

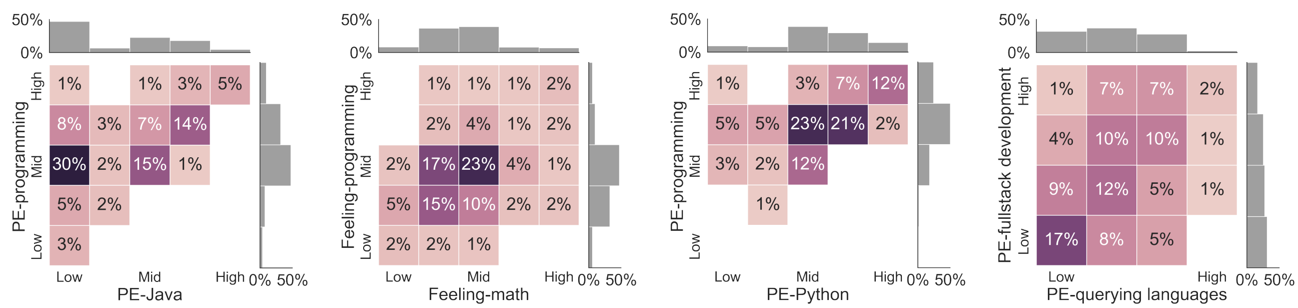
Variable	Tracked in	Text options, from least experienced ("Low" in Figures 1-2) to most experienced ("High" in Figures 1-2)				
PE-prog.	Intro	I have/had never written a line of code in my life.	I have/had played around with coding a bit, but never written anything beyond a few lines. I don't know how to/feel confident writing loops or if statements.		I have/had written multi-line programs, using ifs and loops to build a function or method.	I have/had written many large programs combining multiple functions/methods.
PE-prog.	DSA, Data	No experience (or "Little to no experience")	Some experience, less than a full course	An introductory course	Some experience beyond an introductory course	Extensive experience beyond an introductory course
PE-Java	DSA					
PE-Python	Data					
PE-proof	DM, Algo-Sp23					
PE-full stack dev.	DB	I had no experience with full stack dev.	I had little experience with full stack dev.	I had some experience with full stack dev.	I had extensive full stack dev. experience (one or more internships and/or several courses)	
PE-querying lang.	DB	I had no experience with querying lang.	I had little experience with querying lang.	I had some experience with querying lang.	I had extensive experience with querying lang.	
Feeling-prog.*	Algo (Fa23-Fa24)	Highly underprepared	Slightly underprepared	At the right level of preparedness	Slightly overprepared	Highly overprepared
Feeling-math*	Algo (Fa23-Fa24)					
# stats courses	Data (Sp24-Fa24)	0	1	2	3+	

These are responses to the question: ("How did you feel about being prepared in programming (Feeling-prog.)/mathematically prepared (Feeling-math) for Algo before taking it?"*).



(a) PE-prog. (Intro) **(b) PE-proof (DM)** **(c) PE-proof (Algo-Sp23)** **(d) # stats courses (Data)**

Figure 1: Distribution of PE variables. See Table 2 for complete lists of text options.



(a) DSA **(b) Algo (Fa23-Fa24)** **(c) Data** **(d) DB**

Figure 2: Joint distributions of PE variables. Cells with $< 1\%$ of total students were omitted. See Table 2 for text options.

Help-seeking on class forums. All classes had a class forum on Ed Discussion [8] where students seek asynchronous help by initiating a *thread*. In doing so, students could opt to remain anonymous to their peers or keep the thread *private* (i.e., viewable only to staff). For each student in each class, we analyze the number of threads they read (Reads) over the semester[2] and the number of threads they initiated (Writes). Following Sharma *et al.* [31], we calculated a student's *anonymity rate* (Anon%) as the proportion of the student's *public* threads that are anonymous. Similarly, we define a student's *private rate* (Priv%) as the proportion of their threads that are private. We excluded DM-Sp24 data from Priv% as the instructor disallowed private threads.

Help-seeking in consulting hours. We use *consulting hours* (CH) throughout the paper to refer to a designated time/place (including online) where students could seek *one-on-one* help from a class staff member. All classes in our data hosted regular CH multiple days a week via a queueing app (some used MyDigitalHand [29, 36] and others used an internal tool) that tracked the waiting time and the duration of all interactions. Some classes (bolded in Table 1) also tracked the Outcome of each interaction, where the student could select among *"I did not make any progress", "I will need more*

[2]This is mostly similar to the *lurking* behavior discussed in existing works [31, 38]. The data also tracks the number of days each student logged into the forum (Days), which some related works [20, 38] analyzed. However, we found an extremely strong rank correlation between Days and Reads in our data (with a Spearman's ρ ranging from 0.87 to 0.92 by course). This implies either metric captures students' reading behavior on forums and can serve as a proxy for the other. We thus analyze only Reads.

Table 3: Summary of course-level Jonckheere's trend test results.

Course	Variable	Class Forums									Consulting Hours							
		Reads		Writes		Anon%		Priv%			Requests			MedDuration		Outcome		
		N	Eff.	*N*	Eff.	*N*	Eff.	*N*	Eff.	*p*	*N*	Eff.	*p*	*N*	Eff.	*N*	Eff.	*p*
Intro	PE-prog.	460	+	460	+	198	+	247	+		460	**−0.55**	3.0×10^{-2}	123	+	1086	**+0.54**	3.8×10^{-4}
DSA	PE-prog.	813	−	813	−	324	−	382	−		813	**−0.58**	2.2×10^{-7}	221	+		<No data>	
	PE-Java		−		−		−		−			**−0.57**	2.1×10^{-6}		+		<No data>	
DM	PE-proof	130	−	130	−	72	−	43	−		130	**−0.63**	2.1×10^{-3}	66	−	394	−	
Algo	PE-proof	202	+	202	+	121	+	148	+		202	−		59	−		<No data>	
	Feeling-prog.	433	+	433	−	163	−	193	−		433	−		95	+	897	+	
	Feeling-math		−		−		−		−			**−0.56**	3.1×10^{-3}		−		−	
Data	PE-prog.	402	+	402	+	165	−	223	**+0.59**	0.041	402	−		47	−	428	+	
	PE-Python		−		+		−		+			−			+		**+0.54**	1.8×10^{-2}
	# stats courses	177	+	177	−	72	−	100	−		177	+		13	−	94	+	
DB	PE-full stack dev.	183	−	183	−	106	−	126	−		183	**−0.62**	2.2×10^{-4}	40	−	283	−	
	PE-querying lang.		−		+		−		+			−			+		−	

Plus (+) and minus (−) signs indicate the effect directions (regardless of significance), i.e., students with higher PE are in aggregate more likely to have a higher (+) or lower (−) value of that metric. For brevity, we report the (corrected) p-values and the CLES for significant results only. We omitted the p-value column for metrics without any significant effects.

help", "My problem was not entirely resolved but I can figure out on my own", and "My problem was resolved", though this question was not required. We excluded CH interactions whose wait time exceeded four hours or whose interaction duration was not between 1-60 minutes; these records mainly reflected human errors (staff member not closing the queue after each shift, or interactions that did not take place). Based on the remaining data for each class, we aggregated (1) each student's total number of requested interactions (Requests) and (2) for students with Requests > 3, their median interaction duration (MedDuration).[3]

Analyses. We analyze the relationship between each PE variable and each help-seeking metric *in each course* separately. Following many works in this realm [20, 25, 31, 32, 38, 40, 46], we used non-parametric statistical tests for all metrics.[4]

We use Jonckheere's trend test [16] for all the ordinal PE variables to preserve the ordinal experience levels, and report the common language effect size (CLES). Similar to that of the Mann-Whitney U test, CLES ranges between 0.5 and 1.0, and can be interpreted as the proportion of pairs of students with different levels of PE in which the student with more PE has a higher value of the metric (if the effect is positive) and vice versa. For each help-seeking metric, we conduct all tests at the *course* level (i.e., pooling all offerings of the same course together) and perform Holm-Bonferroni corrections across all courses to control the overall type-I error rate at the 0.05 level. Note that each observation of Outcome is a *CH interaction* while each observation of all other metrics is a student.

We then study the interaction between multiple PE variables. For each course with multiple PE variables, when one or more PE variables exhibit post-correction significant associations with a help-seeking metric, we build a regression model suitable for the metric (see Section 4.3 for details), taking *all* PE variables as input.

[3]We use the median instead of the mean to more robustly capture the length of a typical interaction. We would also note that students' *total* duration rank-correlates almost perfectly with Requests (Spearman's $\rho \geq 0.87$ even with all non-users removed) and thus does not add any meaningful insight to students' help-seeking in CH.

[4]We performed Shapiro-Wilk normality tests on all metrics anyway; *all* tests indicated non-normality with p-values smaller than 0.0002.

4 Results and Discussions

4.1 PE effects in class forums

Table 3 summarizes the results when each PE variable (rows) is tested against each help-seeking metric (columns) in each course. Based on insights from related works, we initially hypothesized that students with less PE would both need more help in general and use external help less effectively [35, 44] than their experienced peers. If so, they would then resort to internal resources [9, 43] more than their peers, resulting in a *negative* effect on both forum usage frequencies (Reads and Writes). However, we observed no significant effects for either metric, and many of the effect directions (see Table 3) turn out to be *positive*, indicating that more experienced students sought marginally more help on class forums in those courses. We also did not observe any significant effects on students' usage of the anonymous features, but instead observed a weak positive relation between PE-prog. and Priv% only in Data.[5]

Discussion. Sibia *et al.* [32] previously observed a negative effect for Anon% (more experienced students using anonymity less) in a large-scale CS1 course that had more high-PE students than low-PE students,[6] but this discrepancy disappeared when they subdivided their forum into smaller subspaces with homogenous PE levels. As mentioned in Section 3, many experienced students skip Intro and directly place into DSA. This diversion likely makes the PE experience gap in both courses smaller (see Figures 1 and 2a), rendering the context in both Intro and DSA more similar to their homogeneous setting. However, the directions of the insignificant effects on Anon% are mostly negative, and we cannot rule out the possibility of a subtle negative effect undetectable in our data.

[5]We initially speculated that this phenomenon stems from Data's unique instructional context: it was the only course that administered take-home exams, during which students were allowed to seek clarification on the class forum *via private posts*, and thus it might be that students with more PE sought comparable clarification during exams (but *less* help otherwise) than students with less PE, which would explain the higher Priv% for students with higher PE. However, upon filtering out all forum threads related to the take-home exams, we still observed the same effect with comparable magnitudes *in all Data classes*, so our speculation does not fully explain the phenomenon.

[6]We note that our results cannot be directly compared to theirs, as their corresponding analyses are at the post level (i.e., each observation is a post, not a student).

Sibia *et al.* [32] also observed that more experienced students had a higher Priv%, and speculated the reason to be that any forum post including solution code has to be private. While this is also true in our context, we did not observe a similar significant effect in either Intro or DSA, and the effect direction was in fact *negative* in DSA. More investigation is necessary to understand the relationship between PE and students' private post usage.

4.2 PE effects in consulting hours

As highlighted in Table 3, we observed significant negative effects between at least one PE and Requests in every course except Data, and non-significant but still negative effect directions for all but one other variable (# stats courses in Data). In other words, more experienced students utilized CH significantly less than their less experienced peers, consistent with our initial hypothesis on help-seeking frequency. We found no significant effects on frequent CH users' MedDuration, suggesting students' PE might not influence the length of a CH interaction once they decide to seek it. On students' experiences in these interactions, we found PE-prog. (in Intro) and PE-Python (in Data) positively associated with Outcome, which indicates that students with more experience had more self-reported progress in CH in these two contexts. However, the effect directions vary for other courses, suggesting that this effect is highly specific to the course context.

Discussion. We discuss two possible explanations for why substantial effects are found in CH but not in the forum. First, CH is arguably a more intimate space than class forums, as most interactions happen directly (and semi-privately) between a student and a teaching staff member. Therefore, inexperienced students might feel less "exposed" and "judged" seeking help in CH than seeking help publicly in forums. Second, CH is more adaptive, more media-rich, and requires less effort from the students to phrase their needs. Since inexperienced students often lack the ability to provide sufficient context for their questions on forums [28, 41, 44], they might gravitate towards CH to actively seek internal help, which would simultaneously strengthen the negative effect in CH and neutralize it in forums. However, more in-depth investigation is needed to untangle these effects and explain why higher PE levels are associated with progress in some contexts but not others.

4.3 Interaction between PE variables

4.3.1 Requests. Most significant results in Section 4.1 involve Requests. We chose to build a hurdle model for each multiple-PE-variable course (DSA, Data, DB, and Algo) for its suitability of modeling count data with many zeros and easier interpretation of parameter estimates [11]. Although Data does not have a significant variable in Table 4, we included them in this analysis to understand the potential interaction between PE variables. Each hurdle model consists of a (binary logit) *zero part* modeling *whether a student is a CH-user* and a *count part* modeling *how many visits each user makes*. We used a truncated negative binomial model for the count part due to overdispersion, as the variance of the nonzero part of Requests is at least 5x its mean in every course.

The results of the hurdle models (summarized in Table 4) collectively show that PE variables are slightly more effective in explaining *whether students use CH at all* (the zero part) than explaining

Table 4: Hurdle model results for Requests.

Course	Variable	Zero part			Count part		
		Coef.	std.	p	Coef.	std.	p
DSA ($N = 813$)	const	0.377	0.194	0.05	1.567	0.470	0.001
	PE-prog.	-0.201	0.068	**0.003***	0.009	0.148	0.95
	PE-Java	-0.123	0.050	**0.014***	-0.197	0.086	**0.021***
Algo ($N = 433$)	const	0.156	0.295	0.53	2.865	0.635	6.5e-6
	Feeling-prog.	-0.099	0.101	0.33	-0.302	0.209	0.15
	Feeling-math	-0.266	0.092	**0.004***	-0.409	0.200	**0.04***
Data ($N = 402$)	const	-0.760	0.512	0.14	-7.081	5.968	0.24
	PE-prog.	0.010	0.144	0.95	0.254	0.291	0.38
	PE-Python	-0.079	0.091	0.39	-0.291	0.157	0.07
DB ($N = 183$)	const	0.298	0.309	0.34	0.562	1.478	0.70
	PE-full stack dev.	-0.445	0.123	**0.0003***	-0.300	0.201	0.14
	PE-querying lang.	0.119	0.146	0.42	-0.052	0.255	0.84

p-values for PE variables significant at the 0.05-level are bolded and followed with an asterisk. Bolded PE variables have larger explanatory power (being significant in both parts or having coefficients with larger magnitude than the other variable in both parts). Dispersion parameters omitted for brevity.

their usage frequency (the count part). Specifically, each PE variable significant for Requests in Table 3 is still significant (with a negative effect) in the zero part of their respective hurdle models. However, not all are significant in the corresponding count parts, and the coefficient for PE-prog. even becomes *positive* for DSA.

Discussion. Each class has one PE variable with more explanatory power than the other (bolded in Table 4). These are also the variables that align more with the course activities than the general course concepts. All DSA assignments are in Java, and all Data assignments are in Python despite the course not having a programming focus. DB's semester-long project requires building a full stack application, rendering PE-full stack dev. more relevant than PE-querying lang., despite the latter being more aligned with the course topic. There are a lot more theoretical problems (algorithm design and analysis) in Algo assignments than applied problems (algorithm implementation and experiments). Thus, Feeling-math has a more pronounced association with student help needs.

Table 5: Ordinal logistic model results for Outcome in Data. Cutoff values for latent variable omitted for brevity.

Variable	Model 1 ($N = 428$)			Model 2 ($N = 94$)		
	Coef.	std.	p	Coef.	std.	p
PE-prog.	0.080	0.214	0.71	-0.544	0.914	0.55
PE-Python	0.270	0.104	**0.009***	0.227	0.458	0.62
# stats courses				0.834	0.574	0.15

4.3.2 Outcome. We then built two ordinal logistic regression models [18] for Outcome in Data. Model 1 (see Table 5) uses all Data classes and takes PE-prog. and PE-Python as inputs, while Model 2 additionally takes # stats courses as an input but only uses Sp24-Fa24 data. Consistent with the result in Section 4.2, Model 1 finds only PE-Python significantly related to Outcome.[7] Although this effect is no longer significant in Model 2, the coefficient for PE-Python is of comparable magnitude (0.23), suggesting that the lack of significance is mainly due to the smaller data size.

[7]Here the model suggests that increasing a student's PE-Python by one level (see Table 2) would lead to an increasing in the *odds* that the student gets more progress (over less progress) by a factor of $e^{0.27} = 1.31$.

Discussion. Among multi-PE variable courses, Data is the only course without any PE variable exhibiting a significant negative effect on Requests in the hurdle models, and also the only course where significant effects on Outcome are observed in Section 4.2. We note that Data also has the lowest mean Requests (1.35) among all courses as well as the lowest percentage of students with nonzero Requests (31%; all other courses are larger than 40%), so it is possible that the sparsity of CH usage supressed the finding of a negative effect in Requests like in other courses. On the other hand, we speculate the Outcome effect might be due to the interdiscplinary nature of Data's course content: students with ample Python experience may be more likely to access CH for conceptual help on statistics and/or data manipulation, which might be easier to address than programming issues (as no debugging is involved). Such CH interactions may then yield more immediate progress than other CH interactions that involved debugging.

5 Limitations and Future Work

Data and scope. Our data comes from a research-oriented institution in the US and may be biased towards student behavior in this context. The data may also be biased towards students who are inclined to give consent to data usage for research, which is not always a representative sample of all enrolled students. All PE data is self-reported by students within limited options (those in Table 2) and may not capture all kinds of relevant experiences. Specifically, none of the courses explicitly tracked *prior help-seeking experience or behavior at the same institution*,[8] which has been reported both qualitatively [24] and quantitatively [20] as affecting students' help-seeking in future courses. Some classes unideally surveyed students' PE in the middle or towards the end of the semester. Although the question wordings all clearly indicated that they asked for students' PE *prior to enrollment in the course*, students' self-assessments could still be influenced by their experience in the current course. Finally, the collection of help-seeking metrics may also be subject to human error (e.g., some CH interaction not logged) or instructional events (e.g., Data disables public posts during online exam periods).

Methodology. Our analyses are mostly carried out at the course level, and thus do not capture potential fluctuations in instructional contexts of different offerings. We made this choice for larger sample sizes/statistical power and easier presentation. However, for all significant results at the course level, we ran class-specific confirmatory tests/models and found that no class-level results meaningfully differed from the course-level aggregate. Our analyses on students' usage of anonymous and private threads are at the student level, and thus cannot be compared directly to thread-level analyses (e.g., *are anonymous posts disproportionately likely to be from less experienced students*) such as that in Sibia *et al.*'s work [32].

Our regression models in Section 4.3 treats PE variables as parametric instead of encoding each into four to five separate dummy variables. We made this choice for model simplicity and explanability. We also note that the purpose of these models is to investigate how/whether each PE variable is significantly associated with help-seeking metrics when all variables are considered together, and not to predict the metrics accurately by using solely PE data. We therefore do not emphasize nor report the model performances, and omit some details in the model.

[8]Some of this may be partially inferrable (e.g., a student took DM in Sp23, sought substantial help from internal resources, then proceeded to take Algo in Fa23), but we leave this for future work.

PE vs. gender. PE is only one of many student characteristics. Other characteristics include gender, race/ethnicity, major, year/standing, and international or first-generation status, for a non-exhaustive list. Although we chose to focus this work on PE and leave all other characteristics (and intersectionality) for a future expansion work with broader scope, the interplay between PE and other characteristics should not be overlooked. Indeed, Sibia *et al.* [32] posited that some effects between PE and students' help-seeking on forums may actually be driven by gender due to the relation between gender and PE in introductory programming contexts. We built a hurdle model for Requests for each course using the same setting in Section 4.3, taking both PE variable(s) and gender as inputs. We found a similar phenomenon to that in Sibia *et al.* [32]'s work in *some but not all* models: the positive effect between PE-prog. and Requests is no longer significant in Intro when gender is explicitly controlled, but many PE variables in other courses remain significant with gender controlled. We leave a more systematic analysis for future work.

Another direction for future work is to broadly study the effects of PE on students' *intentions* of using external help resources (See Skripchuk *et al.* [34] for some results on generative AI and web search) as a complement to internal resources that we focused on.

6 Conclusions and Implications

This work attempts to understand the relationships between relevant PE and student help-seeking behavior (with regards to internal help resources) in computing contexts beyond CS1. We observed that more experienced students seek less one-on-one synchronous help across all courses in our data, with many of these effects being significant. We also found that more experienced students reported making more progress in these help interactions in two courses that used Python. However, students' usage of class forums (including not only their post frequencies but also their rate of anonymous or private posts) did not differ significantly in most of our courses.

Our results highlight the distinction between the user populations of the two kinds of internal help resources we examined, i.e., CH users are more likely to be inexperienced than forum users. It is therefore paramount for teaching staff in large courses to receive adequate training on working with inexperienced students in CH. Moreover, we show that the relationships between PE and help-seeking exist beyond introductory courses, and that general prior programming experience may not be the most relevant type of PE for every computing educator and their instructional context. The unique results observed in Data throughout the paper (see Footnote 5 and Section 4.3.2) also highlight the importance of instructional contexts in student help-seeking.

Acknowledgments

We thank all instructors, teaching associates, and departmental staff for their efforts in help collecting and transferring the data. We thank Janet Jiang, Abby Melton, Alex Pool, and the anonymous reviewers for their helpful and constructive feedback to the paper. This work is supported by NSF grants #1934965 and #2336805.

References

[1] Christine Alvarado, Gustavo Umbelino, and Mia Minnes. 2018. The Persistent Effect of Pre-College Computing Experience on College CS Course Grades. In *ACM SIGCSE TS*. 876–881. https://doi.org/10.1145/3159450.3159508

[2] João Henrique Berssanette and Antonio Carlos de Francisco. 2022. Cognitive Load Theory in the Context of Teaching and Learning Computer Programming: A Systematic Literature Review. *IEEE Trans. Educ.* 65, 3 (2022), 440–449. https://doi.org/10.1109/TE.2021.3127215

[3] Noshaba Bhalli, Vandana Pursnani Janeja, and David Harding. 2024. Effects of Prior Academic Experience in Introductory Level Data Science Course. In *ACM SIGCSE TS*. 1576–1577. https://doi.org/10.1145/3626253.3635505

[4] Nicholas A. Bowman, Lindsay Jarratt, K. C. Culver, and Alberto Maria Segre. 2019. How Prior Programming Experience Affects Students' Pair Programming Experiences and Outcomes. In *ACM ITiCSE*. 170–175. https://doi.org/10.1145/3304221.3319781

[5] Giang Bui, Naaz Sibia, Angela M. Zavaleta Bernuy, Michael Liut, and Andrew Petersen. 2023. Prior Programming Experience: A Persistent Performance Gap in CS1 and CS2. In *ACM SIGCSE TS*. 889–895. https://doi.org/10.1145/3545945.3569752

[6] Melissa Chen, Yinmiao Li, and Eleanor O'Rourke. 2024. Understanding the Reasoning Behind Students' Self-Assessments of Ability in Introductory Computer Science Courses. In *ACM ICER*. 1–13. https://doi.org/10.1145/3632620.3671094

[7] Anael Kuperwajs Cohen, Alannah Oleson, and Amy J. Ko. 2024. Factors Influencing the Social Help-seeking Behavior of Introductory Programming Students in a Competitive University Environment. *ACM Trans. Comput. Educ.* 24, 1 (2024), 11:1–11:27. https://doi.org/10.1145/3639059

[8] Ed Discussion. 2024. https://edstem.org/.

[9] Augie Doebling and Ayaan M. Kazerouni. 2021. Patterns of Academic Help-Seeking in Undergraduate Computing Students. In *ACM Koli Calling*. 13:1–13:10. https://doi.org/10.1145/3488042.3488052

[10] Rodrigo Silva Duran, Jan-Mikael Rybicki, Arto Hellas, and Sanna Suoranta. 2019. Towards a Common Instrument for Measuring Prior Programming Knowledge. In *ACM ITiCSE*. 443–449. https://doi.org/10.1145/3304221.3319755

[11] Cindy Xin Feng. 2021. A comparison of zero-inflated and hurdle models for modeling zero-inflated count data. *Journal of statistical distributions and applications* 8, 1 (2021), 8. http://doi.org/10.1186/s40488-021-00121-4

[12] Carlton J. Fong, Cassandra Gonzales, Christie Hill-Troglin Cox, and Holly B. Shinn. 2023. Academic help-seeking and achievement of postsecondary students: A meta-analytic investigation. *Journal of Educational Psychology* 115 (2023), 1–21. https://doi.org/10.1037/edu0000725

[13] Zhikai Gao, Sarah Heckman, and Collin Lynch. 2022. Who Uses Office Hours?: A Comparison of In-Person and Virtual Office Hours Utilization. In *ACM SIGCSE TS*. 300–306. https://doi.org/10.1145/3478431.3499334

[14] Irene Hou, Sophia Mettille, Owen Man, Zhuo Li, Cynthia Zastudil, and Stephen MacNeil. 2024. The Effects of Generative AI on Computing Students' Help-Seeking Preferences. In *ACM ACE*. 39–48. https://doi.org/10.1145/3636243.3636248

[15] Andrew Jiang and Bogdan Simion. 2022. Help Supports during Online Delivery: Student Perception and Lessons Learnt from an Online CS2. In *ACM SIGCSE TS*. 105–111. https://doi.org/10.1145/3478431.3499369

[16] Aimable Robert Jonckheere. 1954. A distribution-free k-sample test against ordered alternatives. *Biometrika* 41, 1/2 (1954), 133–145.

[17] Michael S. Kirkpatrick and Chris Mayfield. 2017. Evaluating an Alternative CS1 for Students with Prior Programming Experience. In *ACM SIGCSE TS*. 333–338. https://doi.org/10.1145/3017680.3017759

[18] David G. Kleinbaum and Mitchel Klein. 2010. Ordinal logistic regression. In *Logistic regression: A self-learning text*. 463–488.

[19] Shao-Heng Ko and Kristin Stephens-Martinez. 2023. What drives students to office hours: individual differences and similarities. In *ACM SIGCSE TS*. 959–965. https://doi.org/10.1145/3545945.3569777

[20] Shao-Heng Ko and Kristin Stephens-Martinez. 2024. The Trees in the Forest: Characterizing Computing Students' Individual Help-Seeking Approaches. In *ACM ICER*. 343–358. https://doi.org/10.1145/3632620.3671099

[21] Shao-Heng Ko, Kristin Stephens-Martinez, Matthew Zahn, Yesenia Velasco, Lina Battestilli, and Sarah Heckman. 2025. Student Perceptions of the Help Resource Landscape. In *ACM SIGCSE TS*. 596–602. https://doi.org/10.1145/3641554.3701851

[22] Sam Lau and Philip J. Guo. 2023. From "Ban It Till We Understand It" to "Resistance is Futile": How University Programming Instructors Plan to Adapt as More Students Use AI Code Generation and Explanation Tools such as ChatGPT and GitHub Copilot. In *ACM ICER*. 106–121. https://doi.org/10.1145/3568813.3600138

[23] Colleen M. Lewis, Ken Yasuhara, and Ruth E. Anderson. 2011. Deciding to major in computer science: a grounded theory of students' self-assessment of ability. In *ACM ICER*. 3–10. https://doi.org/10.1145/2016911.2016915

[24] Yinmiao Li, Melissa Chen, Ayse Hunt, Haoqi Zhang, and Eleanor O'Rourke. 2024. Exploring the Interplay of Metacognition, Affect, and Behaviors in an Introductory Computer Science Course for Non-Majors. In *ACM ICER*. 27–41. https://doi.org/10.1145/3632620.3671119

[25] Soohyun Nam Liao, Kartik Shah, William G. Griswold, and Leo Porter. 2021. A Quantitative Analysis of Study Habits Among Lower- and Higher-Performing Students in CS1. In *ACM ITiCSE*. 366–372. https://doi.org/10.1145/3430665.3456350

[26] David Liben-Nowell and Anna N. Rafferty. 2022. Student Motivations and Goals for CS1: Themes and Variations. In *ACM SICCSE TS*. 237–243. https://doi.org/10.1145/3478431.3499358

[27] Dastyni Loksa, Lauren E. Margulieux, Brett A. Becker, Michelle Craig, Paul Denny, Raymond Pettit, and James Prather. 2022. Metacognition and Self-Regulation in Programming Education: Theories and Exemplars of Use. *ACM Trans. Comput. Educ.* 22, 4 (2022), 39:1–39:31. https://doi.org/10.1145/3487050

[28] Amogh Mannekote, Mehmet Celepkolu, Aisha Chung Galdo, Kristy Elizabeth Boyer, Maya Israel, Sarah Heckman, and Kristin Stephens-Martinez. 2022. Don't Just Paste Your Stacktrace: Shaping Discussion Forums in Introductory CS Courses. In *ACM SIGCSE TS*. 1164–1164. https://doi.org/10.1145/3478432.3499110

[29] MyDigitalHand. 2024. https://beta.mydigitalhand.org/.

[30] Thomas W. Price, Zhongxiu Liu, Veronica Cateté, and Tiffany Barnes. 2017. Factors Influencing Students' Help-Seeking Behavior while Programming with Human and Computer Tutors. In *ACM ICER*. 127–135. https://doi.org/10.1145/3105726.3106179

[31] Mrinal Sharma, Hayden McTavish, Zimo Peng, Anshul Shah, Vardhan Agarwal, Caroline Sih, Emma Hogan, Ismael Villegas Molina, Adalbert Gerald Soosai Raj, and Kristen Vaccaro. 2023. Engagement and Anonymity in Online Computer Science Course Forums. In *ACM ICER*. 48–62. https://doi.org/10.1145/3568813.3600121

[32] Naaz Sibia, Angela M. Zavaleta Bernuy, Tiana V. Simovic, Chloe Huang, Yinyue Tan, Eunchae Seong, Carolina Nobre, Daniel Zingaro, Michael Liut, and Andrew Petersen. 2024. Exploring the Effects of Grouping by Programming Experience in Q&A Forums. In *ACM ICER*. 206–221. https://doi.org/10.1145/3632620.3671107

[33] Naaz Sibia, Angela M. Zavaleta Bernuy, Joseph Jay Williams, Michael Liut, and Andrew K. Petersen. 2023. Student Usage of Q&A Forums: Signs of Discomfort?. In *ACM ITiCSE*. 33–39. https://doi.org/10.1145/3587102.3588842

[34] James Skripchuk, John Thomas Bacher, and Thomas W. Price. 2024. An Investigation of the Drivers of Novice Programmers' Intentions to Use Web Search and GenAI. In *ACM ICER*. 487–501. https://doi.org/10.1145/3632620.3671112

[35] James Skripchuk, Neil Bennett, Jeffrey Zhang, Eric Li, and Thomas W. Price. 2023. Analysis of Novices' Web-Based Help-Seeking Behavior While Programming. In *ACM SIGCSE TS*. 945–951. https://doi.org/10.1145/3545945.3569852

[36] Aaron J. Smith, Kristy Elizabeth Boyer, Jeffrey Forbes, Sarah Heckman, and Ketan Mayer-Patel. 2017. My Digital Hand: A Tool for Scaling Up One-to-One Peer Teaching in Support of Computer Science Learning. In *ACM SIGCSE TS*. 549–554. https://doi.org/10.1145/3017680.3017800

[37] Anya Tafliovich, Jennifer Campbell, and Andrew Petersen. 2013. A student perspective on prior experience in CS1. In *ACM SIGCSE TS*. 239–244. https://doi.org/10.1145/2445196.2445270

[38] Adrian Thinnyun, Ryan Lenfant, Raymond Pettit, and John R. Hott. 2021. Gender and Engagement in CS Courses on Piazza. In *ACM SIGCSE TS*. 438–444. https://doi.org/10.1145/3408877.3432395

[39] Anna van der Meulen and Efthimia Aivaloglou. 2021. Who Does What? Work Division and Allocation Strategies of Computer Science Student Teams. In *IEEE ICSE-SEET*. 273–282. https://doi.org/10.1109/ICSE-SEET52601.2021.00037

[40] Bogdan Vasilescu, Andrea Capiluppi, and Alexander Serebrenik. 2014. Gender, representation and online participation: A quantitative study. *Interacting with Computers* 26, 5 (2014), 488–511. https://doi.org/10.1093/iwc/iwt047

[41] Mickey Vellukunnel, Philip Sheridan Buffum, Kristy Elizabeth Boyer, Jeffrey Forbes, Sarah Heckman, and Ketan Mayer-Patel. 2017. Deconstructing the Discussion Forum: Student Questions and Computer Science Learning. In *ACM SIGCSE TS*. 603–608. http://doi.org/10.1145/3017680.3017745

[42] Chris Wilcox and Albert Lionelle. 2018. Quantifying the Benefits of Prior Programming Experience in an Introductory Computer Science Course. In *ACM SIGCSE TS*. 80–85. https://doi.org/10.1145/3159450.3159480

[43] Elizabeth Wirtz, Amy Dunford, Edward Berger, Elizabeth Briody, Gireesh Guruprasad, and Ryan Senkpeil. 2018. Resource usage and usefulness: academic help-seeking behaviours of undergraduate engineering students. *Australasian Journal of Engineering Education* 23, 2 (2018), 62–70. http://doi.org/10.1080/22054952.2018.1525889

[44] David Wong-Aitken, Diana Cukierman, and Parmit K. Chilana. 2022. "It Depends on Whether or Not I'm Lucky" How Students in an Introductory Programming Course Discover, Select, and Assess the Utility of Web-Based Resources. In *ACM ITiCSE*. 512–518. https://doi.org/10.1145/3502718.3524751

[45] Stephanie Yang, Hanzhang Zhao, Yudian Xu, Karen Brennan, and Bertrand Schneider. 2024. Debugging with an AI Tutor: Investigating Novice Help-seeking Behaviors and Perceived Learning. In *ACM ICER*. 84–94. https://doi.org/10.1145/3632620.3671092

[46] Matthew Zahn, Lina Battestilli, and Sarah Heckman. 2022. Academic Help Seeking Patterns in Introductory Computer Science Courses. In *ASEE Annual Conference & Exposition*. https://peer.asee.org/41526