Integration of Practical Computing Skills and Co-curricular Activities in the Curriculum

Sara Hooshangi
Virginia Tech
Blacksburg, VA, USA
shoosh@vt.edu

Ryan Buxton
Virginia Tech
Blacksburg, VA, USA
ryanbuxton523@vt.edu

Margaret Ellis
Virginia Tech
Blacksburg, VA, USA
maellis1@vt.edu

ABSTRACT
Participation in co-curricular activities, such as hackathons, coding clubs, and undergraduate research has been shown to have a positive impact on the retention, persistence, and sense of belonging of students in the Computer Science (CS) field. In this paper, we will present the result of a study to assess the impact of integrating co-curricular activities and practical skills into the undergraduate CS curriculum. More than 500 senior CS students were surveyed over a span of four semesters about their comfort level, use of practical skills, and their experience in a sophomore-level required course which was redesigned a few years ago. The new course introduced practical skills such as version control, SQL, command line tools, and web development as a way to better engage the students and prepare them for co-curricular computing experiences. Our data analysis provides insight about when and where students use practical skills, how students feel about co-curricular activities, and the positive impact of the course redesign on the overall student experience.

CCS CONCEPTS
• Social and professional topics → Computing education.

KEYWORDS
Co-curricular activities, Problem-solving skills, Comfort level

1 INTRODUCTION
Student success in introductory CS courses and their subsequent persistence in the computing pipeline has been extensively investigated over the past few decades. These studies range from personal and behavioral factors such as self-regulation, self-efficacy, motivation, and study habits [8, 10, 14, 26, 32], to academic factors such as high school preparation, prior computing experience, and overall student GPA [17]. Furthermore, other factors such as student comfort, social integration, and sense of belonging have also been shown to contribute to the perseverance and success of students in engineering [6, 18] and the computing field [25, 29].

Not all students entering the computing field have had prior exposure to computer science before entering college. Past exposure to such opportunities could strongly contribute to computer science learning, and the mere lack of exposure could place a student at a disadvantage compared to other peers [1, 2]. As we try to address some of the current concerns surrounding recruitment, retention and diversity in the computing field [34], we should work towards bridging the gap between students who have had opportunities to gain experience in the field versus those who lack exposure, resources, and experience [11].

One avenue to expand such opportunities for students is to put a greater emphasis on practical computing skills acquisition and academic co-curricular activities [21, 22]. Co-curricular activities encompass a large array of activities from programming contests, hackathons [30], and CTF (Capture The Flag) competitions [11], to undergraduate research, service learning projects [15], and Computer Science clubs [13, 19, 36]. Such activities promote academic engagement, provide meaningful experiences for students at all levels, enhance a sense of belonging and community building, increase self-confidence/self-efficacy, intersect with experiential learning, and are correlated with retention [7, 9, 11, 12, 14, 31, 33, 35]. Some literature on underrepresented groups indicates that co-curricular experiences have a significant impact on self-efficacy, confidence, and sense of belonging for these groups [7, 31, 37]. In addition, participation in such activities often equipped students with a set of practical skills that are beneficial to their future careers.

2 BACKGROUND AND MOTIVATION
More than a decade ago, with an increased interest in research about the need to improve CS students’ problem-solving skills [16, 27], our department added a sophomore-level problem-solving course as part of the CS undergraduate curriculum. All students majoring in CS are required to take this course, preferably concurrently with our CS2 (introductory data structures) course. The original course emphasized problem-solving heuristics in the context of classic, but more abstract, logic puzzles and math problems. Over the years and through senior exit interviews, it became evident that some students did not find the course useful or relevant, and it was difficult for many to translate the activities presented in this course into skills that could either help them with more advanced CS courses or prepare them for internships or future careers.

Furthermore, our surveys showed that about 25% of sophomore CS students did not engage in any computing activities outside of their coursework, and the percentage was much higher among
Students who do not participate in co-curricular computing activities have less experience and comfort with informal learning environments and new technologies, and as a result of that, their persistence in the pipeline might be hindered [2], or they may be at a disadvantage when looking for internships and jobs. Practical skills such as the use of the command line, git and version control, and designing and deploying a website are not necessarily core components of an undergraduate curriculum. However, students are expected to employ such practical skills in upper-level CS classes, internships, and undergraduate research.

Motivated by such observations and to address these gaps, our department decided to infuse practical skills within the CS undergraduate curriculum by redesigning the above-mentioned problem-solving course. Such an approach could provide the benefits of co-curricular activities to all students and encourage more participation in co-curricular activities. The redesign of this course would allow students to acquire practical technical skills at an earlier stage in their studies, experience tinkering, build soft skills, and possibly increase participation in co-curricular activities.

Many institutions make efforts in this direction with one-hour courses on practical skills such as UNIX, SQL and Docker. Some even have a class specifically for practicing coding interview questions. Some CS0 courses and the Advanced Placement Computer Science Principles course attempt to expose introductory students to the breadth of computing at a less-technical level. At MIT, a course was created in 2020 as the ‘missing-semester’ to help bridge this recognized gap in practical computing skills for CS undergraduates [28]. Our study helps further understand students’ use of practical computing skills and their feelings about co-curricular activities. We also investigated whether course updates made an impact on students’ experiences and perceptions.

3 CONCEPTUAL FRAMEWORK AND RESEARCH QUESTIONS

This study is guided by Lent, Brown, and Hackett’s Social Cognitive Career Theory (SCCT) [23, 24] and Astin’s Student Involvement Theory [3] to better understand the context in which students use practical skills inside or outside of the classroom.

Social Cognitive Career Theory (SCCT), rooted in Bandura’s general social cognitive theory and self-efficacy [4, 5] provides a framework to understand the different aspects of career development from the formation of career-related interests and subsequent educational and career choices, to the eventual performance and persistence in those careers. In this framework, self-efficacy serves as a primary motivator that predicts career interests, choices, and performance. How self-efficacy, expected outcome, and goal mechanisms interrelate with other personal characteristics (e.g., gender), contextual (e.g., support system), and experiential/learning factors [23] is also examined in this framework.

Student Involvement Theory considers both the quality and quantity of physical and psychological energy that students invest in their academic experience [3]. The involvement ranges from the level of engagement with one’s academic work to participation in extracurricular activities and social interactions with peers, faculty and staff at the institution. According to this theory, there is a correlation between student involvement and student personal development and academic performance.

Guided by these two theories, we aim to examine the effect of incorporating practical skills and elements of co-curricular activities into the curriculum (through a course redesign) and the potential impact on student academic success, preparation for future careers, and comfort level with co-curricular activities. Our research questions are as follows:

- RQ1: Where do students use practical skills?
- RQ2: Did the course redesign impact students’ perception of their academic and non-academic preparation?
- RQ3: What are the students’ comfort levels with co-curricular activities and did the redesign of the course change their feeling in that regard?

4 INSTITUTIONAL BACKGROUND

Virginia Tech is a large research university located in the southeast United States with more than 30,000 undergraduate students and a rapidly growing CS department. The current CS undergraduate enrollment is around 1300 students, and about 500 students graduated with an undergraduate degree in Computer Science in the Spring of 2021. Computer Science is one of 13 departments within the College of Engineering. Students are admitted to the College of Engineering as General Engineering majors in their first year at the university, and they typically declare a major at the end of their freshman year.

4.1 The course

All students majoring in Computer Science are required to take a sophomore-level problem-solving course. The redesign of the problem-solving course is described in our previous work [11, 12] and was primarily focused on bringing practical skills and co-curricular type activities into this course as a way to integrate such skills as part of the curriculum. The intent of the course redesign was to expose students to a wide variety of CS topics to build their practical computing skills within their required coursework. One of the authors of this paper spearheaded the redesign, coordinates the course, and was one of several instructors for three of the past seven semesters. There are usually 4 sections of 40 to 80 students per semester.

Topics were chosen that could be useful during internships, undergraduate research, upper-level courses, and co-curricular activities such as hackathons and coding clubs. The redesign was first implemented in the Fall of 2018 and exposed sophomore-level students to contemporary computing skills across various areas of computer science [12]. Practical skills such as command line, git, version control, wireshark, and use of Python and SQL to explore data science concepts were incorporated into the new version of the course. Students also participated in Capture the Flag type exercises and developed websites as a way to introduce them to these kinds of co-curricular activities. The intention of including such activities was to provide an opportunity for students to practice these applied skills collaboratively in a team, in a comfortable environment, and with appropriate support throughout, thus improving the equity of
experience for students who had and had not previously used the

4.2 Survey Data

Over the past three years, senior CS students in our department have participated in a survey to reflect on their experience in the problem-solving course that they took as sophomores. The survey, administered during a senior seminar series that all senior CS students take part in, includes multiple choice, Likert-type, and open-ended questions where students report on how the skills introduced in that class influence their overall computing experience. Surveys have been conducted during 4 semesters, with more than 550 student responses. Since the redesign of the course happened in the Fall 2018 semester, the first two semesters of the senior survey data included many students who had taken the course pre-redesign. A question on the survey specifically asked students if they took the course before or in/after the Fall 2018. As summarized in Table 1, 38% (209) of students took the course before the redesign and 62% (341) took it after the redesign in or after Fall 2018 semester. Having a sufficient number of students in these two categories allows us to compare the responses of the students who took the original version versus those who took it after the redesign. This project has an approved IRB from our office of institutional research.

Table 1 shows the distribution of the responses based on gender and race for students who completed the survey both before and after the redesign. We limited our data to those survey respondents who had completed the problem-solving course at our institution and excluded data from students who indicated that they did not complete this course at our institution. This inevitability excludes transfer students who received transfer credits for this course, and the implication of this is discussed in the Threats to Validity section. For demographic categories, we have used a similar grouping as in other Computing Research Association (CRA) publications, but our data does not separate non-resident students from domestic students. To protect the anonymity of students in categories with small sample sizes (as required by our IRB guideline), we have combined a few groups together. In our dataset, URM (Underrepresented Minority) includes the following categories; Black and African American, Hispanic, Native American, and multi-racial students. Students without a disclosed race are marked as Not Reported. For gender, the two categories of Prefer to self-identify and Prefer not to respond are grouped together and labeled Other/NR.

Survey questions were designed based on student interest and perception of course usefulness as these factors are known to influence students’ motivation [14, 20]. We surveyed the students about all the skills used in the course but decided to only analyze and discuss the ones that were the most useful outside of class, other than Python which we disregard in this paper since it is a required language at many other institutions. Some students don’t participate in co-curricular activities for a variety of reasons, such as not having enough time, so we were interested in students’ comfort with co-curricular activities to help gauge their preparation and perception. As recently reaffirmed by George et al., students’ psychosocial attributes are important for retention [14]. We know from a previous study that a subset of our CS majors feel intimidated, or apprehensive and not comfortable, and hope to improve such students’ experiences.

5 RESULTS

In this section, we will address each research question by analyzing survey data used to formulate the research questions for this project.

5.1 RQ1: Where do students use practical skills?

In order to get a better sense of the use of practical skills inside and outside of the classroom, one of the survey questions asked students to indicate whether or not they had used a set of practical skills in the following five settings: courses, internships, personal projects, undergraduate research, and competitions. The chosen practical skills we focused on were git/version control, command line, web development, and SQL. Students use UNIX and command line skills in their three required systems courses and exposure in the problem solving course is intended to ease the transition. The other skills were chosen to include topics that were not covered in required courses in our CS curriculum until the redesign of the problem-solving course, but those that are commonly understood to be essential for student success in their careers. For example, git and version control were not taught specifically in our curriculum, but are used on daily basis in software development internships or entry level positions.

Figure 1 shows that students use these skills abundantly in courses, internships and personal projects, and the more prominently used skills in all settings are command line and git/version control. This result also indicates that while a large number of students are participating in internships and are engaged in personal projects, a significantly lower number of students are participating in co-curricular activities such as competitions or undergraduate research.

In order to examine the effect of the course redesign, we compare before and after the course redesign responses for each of the settings described above as shown in Table 2. Our analysis reveals that a similar pattern of skill usage is seen for our students, regardless of the version of the problem-solving course they took. The only exception is in the personal projects setting, where there is a higher percentage of students reported using the skills after the redesign.

This indicates that by the time students are seniors, they are learning and have seen these practical skills somewhere along the way, even if it is not in a required course in the CS curriculum.
However, a higher percentage of the students who took the updated version of the course, which exposed them to practical computing skills, used each of the skills in personal projects. For the students who took the new version, 14% more of them used the command line, 10% more used git, 4% more used SQL, and 9% more used web development skills in personal projects. This means early exposure to practical computing skills in a required course had an impact on student personal projects.

5.2 RQ2: Did the course redesign impact students’ perception of their academic and non-academic preparation?

Senior students taking our survey were asked to rate their experience in the problem-solving course on a five point Likert scale (1-Strongly Disagree, 2-Disagree, 3-Neither Agree nor Disagree, 4-Agree, 5-Strongly Agree) by answering several questions. The questions are grouped according to the following categories: interest in the course, perceived improvement of technical skills, preparation in subsequent CS courses, preparation for career success, and impact on computing activities outside of the classroom. The result is shown in Figure 2, where we compare average response ratings for each category, for students who took the course before and after the Fall 2018 redesign.

The perception of the course varied according to gender, shown in Table 4. Aside from finding the course interesting, the female students perceived that the previous version of the course had less impact on them than the male students. They also found the updated version of the course more interesting than the male students did. A two-sample Welch’s t-test analysis, shown in Table 4, determined that for female students, there was a significant difference in students’ perception and use of practical skills before and after the redesign across each of the areas. For male students, the significant difference was seen in course interest, technical skills improvement, and career preparation. Because of the small sample size in our URM group, a similar analysis on demographics did not provide meaningful insight.

5.3 RQ3: What are the students’ comfort levels with co-curricular activities and did the redesign of the course change their feeling in that regard?

The course redesign slightly improved students’ comfort with co-curricular activities. Students reported their feelings about a range...
Table 4: Student experience before & after; by gender

<table>
<thead>
<tr>
<th></th>
<th>Before</th>
<th>After</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Male</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Found course interesting</td>
<td>3.1</td>
<td>3.5</td>
<td>.002**</td>
</tr>
<tr>
<td>Improved technical skills</td>
<td>2.9</td>
<td>3.1</td>
<td>.03*</td>
</tr>
<tr>
<td>Prepared for future courses</td>
<td>2.8</td>
<td>3.0</td>
<td>0.1</td>
</tr>
<tr>
<td>Prepared for career success</td>
<td>2.8</td>
<td>3.1</td>
<td>.048*</td>
</tr>
<tr>
<td>Influenced co-curricular activities</td>
<td>2.7</td>
<td>2.9</td>
<td>.08</td>
</tr>
<tr>
<td><strong>Female</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Found course interesting</td>
<td>3.2</td>
<td>3.7</td>
<td>.03*</td>
</tr>
<tr>
<td>Improved technical skills</td>
<td>2.7</td>
<td>3.4</td>
<td>&lt;.001***</td>
</tr>
<tr>
<td>Prepared for future courses</td>
<td>2.7</td>
<td>3.3</td>
<td>.003**</td>
</tr>
<tr>
<td>Prepared for career success</td>
<td>2.7</td>
<td>3.3</td>
<td>.005**</td>
</tr>
<tr>
<td>Influenced co-curricular activities</td>
<td>2.5</td>
<td>3.1</td>
<td>.003**</td>
</tr>
</tbody>
</table>

of co-curricular activities on a scale of 1-Very Apprehensive, 2-Somewhat Apprehensive, 3-Somewhat Comfortable, 4-Very Comfortable. As shown in Figure 3, the average comfort level reported was higher for students who took the updated version of the course. The largest increase was for personal projects and the smallest increase was for undergraduate research. For the overall dataset, there are no significant differences for each activity between before and after the course update based on a two-sample t-test. Likewise, there are no significant differences in reported comfort-levels between before and after based on race or gender.

Figure 3: Student comfort level before and after redesign

However, as shown in Table 5, there is a significant difference in the comfort level for personal projects and online challenges between overall male and female students based on a Welch two-sample t-test, but no significant difference was observed based on race.

Table 5: Student comfort level by gender

<table>
<thead>
<tr>
<th></th>
<th>Male</th>
<th>Female</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Personal Projects</td>
<td>3.5</td>
<td>3.3</td>
<td>.005**</td>
</tr>
<tr>
<td>Online Coding Chall*</td>
<td>2.9</td>
<td>2.7</td>
<td>.01*</td>
</tr>
<tr>
<td>UG Research ‡</td>
<td>2.7</td>
<td>2.8</td>
<td>.36</td>
</tr>
<tr>
<td>Competitions</td>
<td>2.5</td>
<td>2.4</td>
<td>.06</td>
</tr>
</tbody>
</table>

*Chall: Challenges; ‡ UG: Undergraduate

Figure 4 shows that, in general, students feel the most comfortable with personal projects and somewhat comfortable with online coding challenges. Overall, students are more apprehensive than comfortable with undergraduate research and competitions. Male and URM groups follow the same pattern, but interestingly, female students feel better about undergraduate research.

5.4 Response to an open-ended question

Our survey, in addition to the quantitative questions, also included an open-ended question where students could describe how the problem-solving course influenced their computer science activities outside of the classroom. The majority of the responses received for this question were from students who took the course after the redesign, and they consistently emphasized the positive impact of the course on their engagement with other activities. Different students stated the benefits of different parts of the course as it related to participation in competitions, being more active with personal projects, or the use of the practical skills taught in the course in an internship. Table 6 provides a sample of student comments (for students who took the course after the redesign).

6 DISCUSSION

In this work, we analyzed survey data collected in a study, with the aim of better understanding the context in which students use practical skills and co-curricular activities inside or outside of the classroom. Our findings showed that students use practical skills primarily in courses, internships, and personal projects, but not as many participate in the more traditional co-curricular activities such as coding competitions, hackathons, or undergraduate research. Student participation is consistent before and after the redesign with the exception of personal projects, where a higher percentage of students reported being engaged in such activities after the redesign. This suggests early exposure to practical computing skills in a required course had an impact on student personal projects, but not necessarily on other co-curricular activities. Given the positive impact and benefit of participating in co-curricular activities on academic engagement, enhancement of a sense of belonging, and community building for students, [11, 12, 31], we need to do a better job of promoting such activities.

We further examined the effect of incorporating practical skills and elements of co-curricular activities into the curriculum and compared student perception, experience, and comfort level before
and after the curriculum change. Student perception is positive about the course updates, but more targeted course changes and further study could be pursued to make a more substantial impact on students’ feelings about pursuing co-curricular activities. In particular, course changes to scaffold student experiences with personal projects and online coding activities could improve comfort levels for all students and especially female students. These are two components that potential employers often refer to during the hiring process and could impact job offers, career opportunities, and salary.

Because the course subtly exposes students to many topics and addresses problem-solving/practical computing skills, and exposure to co-curricular activities, it may need a more targeted objective to have a stronger impact in a given area. There may also be other instructional design approaches to improve student experience or curriculum design approaches to target which students would benefit from exposure to certain skills and co-curricular activities.

Future studies can analyze categories of student responses and also perform a factor analysis. The open-ended responses from both the senior survey and the post course survey can be analyzed qualitatively, and students can be interviewed to answer the remaining research questions. Pre and Post surveys of the course can be analyzed to further understand the reasons students don’t participate in co-curricular activities and the overall impact of previous computing experience on students’ practical computing skills. The impact of this course can also be measured and tracked in required upper-level undergraduate courses.

7 THREATS TO VALIDITY

Internal Validity: Our sample size is large, and we believe the diversity in our student population is enough so that our study does not suffer from selection bias, however, the exclusion of transfer students who didn’t take the course skews the demographic makeup of our senior class. Differential experience could be a potential threat to this study, as students come from diverse academic backgrounds. All of these students took the survey during their senior year, but there could be many paths the students took to get to that point in their academic career and the variation in experiences could affect where they use practical computing skills, their perception of their academic preparation and their feelings about co-curricular activities. External Validity: Our findings are based on a particular course at a larger research institution. As such our results might not be generalizable to all undergraduate students. However, the characteristics of our students are representative of typical undergraduate students majoring in CS at large research institutions in the United States.

8 CONCLUSION

We determined where and to what extent our students used various practical computing skills and students’ overall comfort with various co-curricular activities. Command line and git/version control were the most commonly used reported skills in internships, personal projects, undergraduate research and competitions while the use of SQL and website development varied. Overall, students were most comfortable with personal projects out of the co-curricular activities studied, followed by online coding challenges and undergraduate research. Students were most apprehensive about competitions. Female students reported being significantly less comfortable with personal projects and online coding challenges than their male counterparts. The updates to the course did not have a statistically significant impact on students’ apprehension about co-curricular activities.

The updates to the problem solving course did have a statistically significant improvement on students’ perception of the course and its usefulness. Students found the updated version of the course more interesting and they reported improving their technical skills more in the updated version of the course. Although on average the course didn’t influence students’ co-curricular activities, the updated course improved this perception. Students also reported that the updated course better prepared them for their subsequent courses and future careers.
REFERENCES


