An Alternative to Programming Contests*

Timothy Urness
Department of Mathematics and Computer Science
Drake University
Des Moines, IA 50311
timothy.urness@drake.edu

Abstract

Programming contests, such as the International Collegiate Programming Contest (ICPC) have many benefits for undergraduate students studying computer science. These include encouraging collaboration amongst teammates and providing an incentive to study of algorithms and data structures. However, these multi-hour, time-pressured contests also have the potential to discourage students and promote bad habits in software development (e.g. rushed development, non-reusable code, and the lack of documentation). In this paper, we describe an alternative to traditional programming contests that involve student-led projects that span an entire semester. These projects, which include long-term data science competitions and student-led research projects, have the benefit of exploring the cutting-edge of technology while giving students the opportunity to collaborate, learn from mistakes, and develop robust software that incorporates aspects of software engineering.

1 Introduction

Programming contests, such as the International Collegiate Programming Contest (ICPC) have been a popular activity amongst undergraduate computer science programs. The contests can benefit students in many ways. The ICPC website (https://icpc.baylor.edu/regionals/abouticpc) states:

> "The contest fosters creativity, teamwork, and innovation in building new software programs, and enables students to test their ability

> to perform under pressure. The contest has raised aspirations and performance of generations of the world's problem solvers in the computing sciences and engineering."

We have seen, firsthand, many of these benefits described above. The contest provides an atmosphere where students are excited to compete and debrief after the contest has concluded. The contest itself can motivate students to join a club to develop skills and deepen their knowledge of computer science algorithms and data structures in order to be more successful in future competitions.

However, we have also witnessed situations where students leave the contest dejected for a number of reasons. Primarily, students can be frustrated that code would pass all of the examples provided, but not pass all of the hidden testing examples. The contest rules stipulate the ranges of boundary cases that are not thoroughly supplied in the testing examples provided. While this is an important component of the contest, the feedback given to the students is limited. This black-box testing leaves little room for constructive feedback and has caused students to leave the contest feeling frustrated, inadequate, and under-educated. Instead, a more important skill that we would like to encourage is the dedication to utilize available resources to find a solution after a problem is encountered. The short-term nature of the competition has the potential to discourage students that would otherwise thrive if time and resources were available.

Another practice that is unintentionally encouraged by time-pressured programming contests is the development of hastily-developed code designed to run only once. Much of the code that is constructed in programming contests does not adhere to well-accepted software engineering practices (e.g. documentation, object-oriented programming, reusable code). Short-term contests do not necessarily promote the skills of software developers that computer science professors are often encouraging thorough the curriculum.

Lastly, IBM has recently discontinued as the ICPC corporate sponsor. At the 2018 and 2019 North Central North America Regional contest, participation required a $25 fee per participant. The fees may prove to be a burden to departments wishing to open the contests to all interested students.

2 Related Work

There have been other alternatives to programming contests proposed. Many of these references make a point that the ICPC stresses skills that are not always in line with modern computer science and software engineering degree programs. Competitions can play a productive role in education and competitive desires can be utilized for educational motivation [4], though competition in education can be good or bad [3]. The best results are often obtained when competition is combined with cooperation. Furthermore, competitions should not be the only factor in assessment and is best utilized as a complement to standard teaching [3].

The College of Charleston student chapter of the ACM hosted a contest in which the judging criteria included both technical and artistic merit[1]. The contest consisted of problems that included some working scaffolding code and a *syntax master expert*, who could answer any student question about syntax. Furthermore, the problem statements encourage a design phase, an implementation phase, a testing phase, and a final submission. In addition to technical correctness, the judging rubric included several sections that evaluated the quality of the submitted code, testing cases, and results.

Constantinescu et. al describes a contest that contains a component aimed at boosting students' creativity and involves a presentation of a finished project to the judges [2]. A high-scoring balanced solution considers multiple facets of the contestants' talent, effort, and results.

In this paper, we provide another alternative to a multi-hour, time-pressured programming contest.

3 An Alternative to the Programming Contest

As an alternative to the programming contest, we've developed semester-long, student-led research groups that have faculty mentors. The groups are designed to be a place where students can explore exciting and cutting-edge aspects of computer science and develop skills over the course of several months. The groups are student-led where faculty involvement can be minimal and is more akin to a mentor or coach than a research director.

The groups are not associated with any credit-bearing course or internship. Instead, we incentivize students by stressing the fun nature of exploring the cutting-edge of technology, the collegiality of being involved with a group, and the potential to contribute to a published research paper. In addition, we stress that participation in a group has the strong potential to increase the strength and warmth of a recommendation letter that a professor could write for a student.

3.1 Forming Groups

During the first week of the semester, we invite all students studying computer science, mathematics, or data analytics to an informative meeting to introduce the idea of student-led research groups. We present many possible ideas for areas of study. In the past, project ideas have included: virtual reality, game development, robotics, mathematical modeling, machine learning, computer science theory, hardware, 3D printing, data science competitions, and data mining. These topics align roughly with faculty interests, but also include topics at the suggestions of students. After the initial meeting, we have students separate into smaller groups of common interests.

3.2 Student-Led Projects

As part of smaller groups, students are tasked with developing concrete goals for the semester. This critical juncture of the projects may require the most professor guidance as the specific goals will determine the long-term success of a project. For example, the transition to a vague interest in a topic such as virtual reality to a specific project (e.g. creating a VR museum) has the potential to cause students to sustain interest or dissuade involvement based on the individual preferences. Students in the smaller groups also determine the best time to meet and accomplish short-term goals. The short-term goals often times include identifying tutorials, development environments, and small tasks to bring to the larger group.

3.3 Online Competitions

Another option for the research groups is to work on an online data science competition. Table 1 shows several examples of data science competitions from kaggle.com that were open in fall of 2019. The prize money is likely out of range for an undergraduate team, particularly with little experience, but the potential of competing for prizes may be the positive motivation that drives students to learn more [4].

Table 1: Fall 2019 kaggle.com Contest Examples

Title	**Description**	**Prize Money**
NFL Big Data Bowl	How many yards will an NFL player gain after receiving a handoff?	$75,000
ASHRAE - Great Energy Predictor III	How much energy will a building consume?	$25,000
Understanding Clouds from Satellite Images	Can you classify cloud structures from satellites?	$10,000
House Prices: Advanced Regression Techniques	Predict sales prices and practice feature engineering, RFs, and gradient boosting	Training Set Only

The learning curve for these competitions is rather gradual as most competitions on kaggle.com also come with introductory tutorials. A group of students can follow a tutorial that will allow them to make a submission to the competition. Then, they can continue to work, refine the algorithm, learn new techniques and re-submit and see if they are able to climb the leaderboard. This long-term learning process rewards collaboration, elaborate solutions, and the exploration of current technology.

There are many other online competitions that groups of students or individuals can utilize to the same effect as kaggle.com. For example, Hacker Rank (https://www.hackerrank.com) provides competitive programming challenges. Programmers are ranked on a leaderboard and can earn badges based on accomplishments. Another online site that provides interesting problems that could be solved in a similar fashion to a programming contest is Project Euler (https://projecteuler.net).

3.4 Goals

The goals for the computer science department at Drake University include providing opportunities for students to participate in beyond-the-classroom projects. We feel that this provides learning opportunities that can complement the traditional curriculum and develops recognition for the program. Concepts such as team-programming, self-motivated learners, code repositories, using code libraries, and learning new concepts by exploring and synthesizing online content are skills that students will need to develop as part of today's technological industry. We feel that students' educations can be enhanced with more dedicated practice with these important skills and concepts, both in and out of the classroom.

Furthermore, if the department can establish an environment where out-of-classroom learning becomes the norm, the research groups can be self-sustaining and produce outcomes after the students that started a particular group have graduated.

3.5 The Role of the Professor

One of the main motivating factors for starting student-led projects is to allow any student, regardless of their experience, to contribute to a research project. In the past, professors would typically select the top few students from a class to join a research project which they directed. If a student was not fortunate enough to get the attention of a professor, his or her education might lack this enrichment opportunity. To give all students an opportunity, we wanted to provide an experience that we could honestly market to all prospective students and not just the students with the potential to be amongst the top few percent.

However, opening up research opportunities to all students has the potential to monopolize a willing professor's time and resources. In order to mitigate this, we set up the initial meetings and opportunities to take place within an hour on Friday afternoons. This is when willing professors will make themselves available for consultation on various projects.

We intentionally have named the projects "student-led" projects and remind students that the main thrust of motivation must come from the students, and not the professor. In this context, professors are seen as coaches and mentors, and not the driving force of motivation behind the projects [5].

4 A Case Study

4.1 Semester-Long Research Group and a Programming Contest Group

During the academic year of 2018-2019, a group of students at Drake University participated in both the student-led research groups. A different group of students participated in the ICPC programming contest in the fall.

The research group met approximately one to two hours a week either independently or with a faculty member over the course of both semesters. The programming contest team only met once or twice prior to the programming contest event held in the fall.

We had over 40 students participate in student research groups and 9 students participate in the programming contest. At the end of the academic year, we asked both groups to anonymously answer two questions on a Likert-like scale from 1 to 5 to assess the student satisfaction with the experience as well as the student's likelihood to participate against next year. The survey also allowed for students to give optional written feedback. The questions posed to the students were as follows:

- How satisfied were you in participating in the event (programming contest or the research group)?
- If you are able, how likely are you to participate next year?
- (optional) Do you have any additional comments?

Twelve students responded for the research group survey, and five students responded for the programming contest survey. Valuable insights into the effectiveness of both groups is available. The results of the satisfaction question are shown in Table 2 and Figure 1. The results of the likelihood of participating in the next year are shown in Table 3 and Figure 2.

Table 2: Satisfaction Survey Results

Group	1 (not satisfied)	2	3	4	5 (very satisfied)
Research Projects	0%	0%	33%	33%	33%
Programming Contest	0%	0%	20%	40%	40%

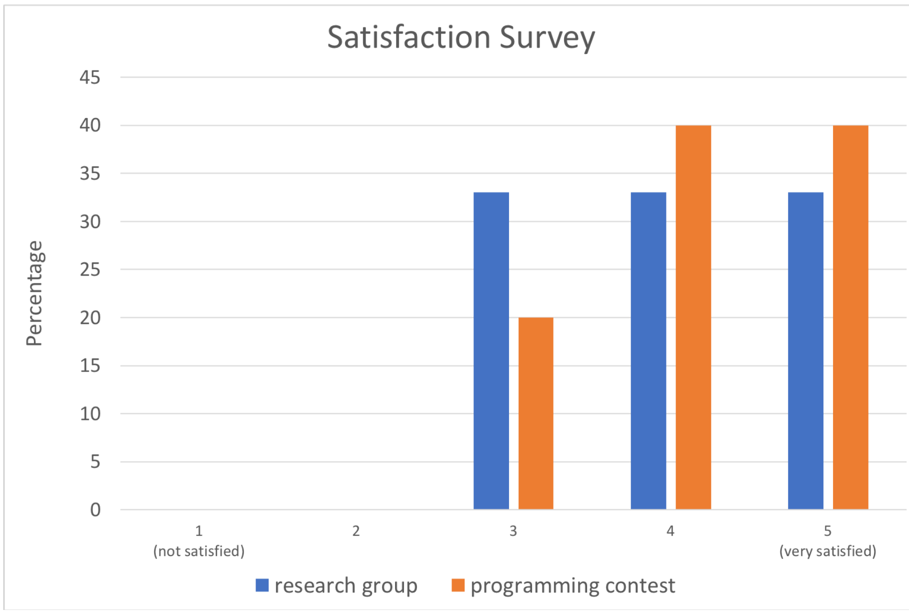

Figure 1: Visualization of the data displayed in Table 2. Answers to the question, "How satisfied were you in participating in the event?"

4.2 Observations and Insights

In aggregate, both groups of students had positive experiences. Each student that completed the survey indicated that there were satisfied with the experience, indicating their satisfaction at a level of 3 (satisfied), 4, or 5 (very satisfied) out of a 5-point scale. A vast majority of the students also indicated that they were likely or very likely (5 out of 5) to participate in a group again in the next academic year. The survey also allowed for a few deeper insights.

The student-led research groups had a higher percentage (66.7% vs 40%) of students rate their likelihood of participating again next year as "very likely" (5 out of 5). However, there were also a few students that rated their likelihood in participating in the research experience next year as "not likely" (1 out of 5) and another student rate their satisfaction as 2 out of 5. The open-ended comments in the survey indicated that at least one student felt that the research experience could be improved by better articulating short, medium, and long-

Table 3: Likelihood of Returning Survey Results

Group	1 (not likely)	2	3	4	5 (very likely)
Research Projects	8.3%	8.3%	8.3%	8.3%	66.7%
Programming Contest	0%	0%	0%	60%	40%

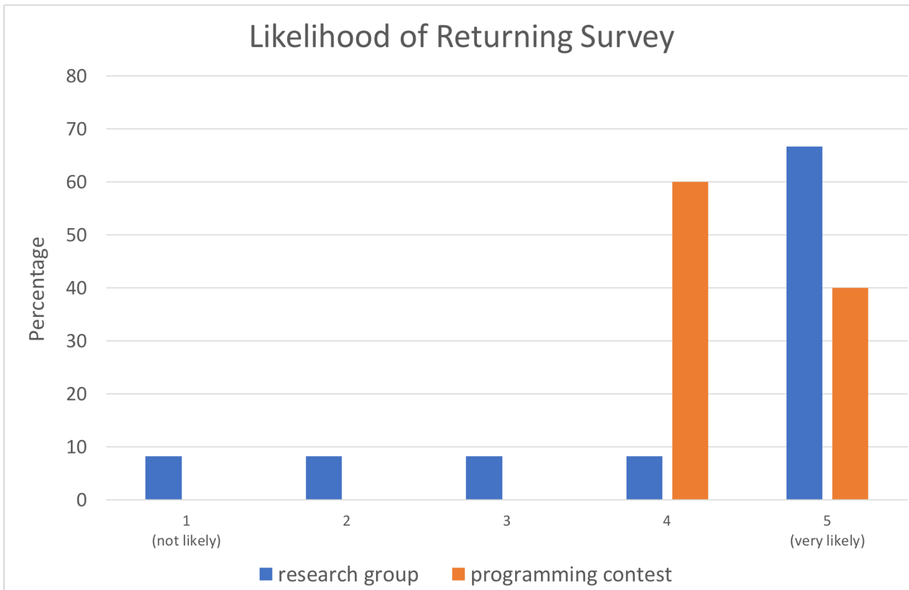

Figure 2: Likelihood of Returning Survey. Visualization of the data displayed in Table 3. Answers to the question, "If you are able, how likely are you to participate next year?"

range goals for each of the students. As the participation in these projects does not hold external incentives in the form of college credit or grades, making the other incentives (e.g. student camaraderie, future letters of recommendation, exploring the cutting-edge of a discipline) may need to be emphasized.

The results from the programming contest survey were surprisingly positive. After the contest, students' reactions were initially mixed. On the ride home, student verbally commented about their frustration with the level of feedback allowed, the length of the multi-hour contest, and the contest structure itself. However, those that responded to the survey indicated they were largely satisfied and likely to participate again next year. We feel that the programming contest team experience could also be improved by treating the group as a club, with regularly-scheduled meetings, practice sessions, and mentoring opportunities designed to mitigate the potentially negative characteristics of a programming contest identified in section 1.

5 Conclusion

In this paper, we've described an alternative to multi-hour, time-pressured programming contests by providing students with opportunities to get involved in semester-long student-led research groups. These groups have advantages over programming contests as they can promote the skills of software developers, facilitate discovery of new concepts not introduced in standard courses, and still provide a competition that can motivate students. A survey of students that participated in either the research groups or the programming contest at Drake University in 2018-2019 found that students found both kinds of extracurricular involvement satisfying, and a vast majority of those that responded will plan to do the events in the future. The student-led research groups had a higher percentage of students rate their likelihood of participating again next year as "very likely" (5 out of 5), but also had a few students that will likely not participate in the future.

Overall, the student-led research groups provide an alternative, and not necessarily a replacement, to the traditional programming contest for some students. These projects, which include long-term data science competitions and student-led research projects, have the benefit of exploring the cutting-edge of technology while giving students the opportunity to collaborate, learn from mistakes, and develop robust software that incorporates aspects of software engineering.

References

[1] James F. Bowring. A new paradigm for programming competitions. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '08, pages 87–91, New York, NY, USA, 2008. ACM.

[2] Zoran Constantinescu, Simona Nicoara, Monica Vladoiu, and Gabriela Moise. Computer science student contests: individuals or teams. In *Proceedings of the 16th RoEduNet International Conference: Networking in Education and Research, Petru-Maior University of Targu Mures, Romania*, 2017.

[3] Fredrik Kristensen, Olof Troeng, Mohammadhassan Safavi, and Prakash Narayanan. *Competition in higher education–good or bad?* Lund University, 2016.

[4] Tom Verhoeff. The role of competitions in education. *Future world: Educating for the 21st century*, pages 1–10, 1997.

[5] Matthew Zwier and Timothy Urness. Just in time research: The advantages and pitfalls of a student-led interdisciplinary undergraduate research experience. *J. Comput. Sci. Coll.*, 33(5):179–185, May 2018.